# PENERAPAN SISTEM KRIPTOGRAFI ELGAMAL ATAS $\mathbb{Z}_p{}^*$ DALAM PEMBUATAN TANDA TANGAN DIGITAL

**SKRIPSI**

Diajukan Kepada Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Negeri Yogyakarta
Untuk Memenuhi Sebagian Persyaratan
Guna Memperoleh Gelar Sarjana Sains


**Oleh :**
**Rininda Ulfa Arizka**
**07305141010**


**PROGRAM STUDI MATEMATIKA**
**JURUSAN PENDIDIKAN MATEMATIKA**
**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**
**UNIVERSITAS NEGERI YOGYAKARTA**
**2011**

**PERSETUJUAN**

**Skripsi**

**Penerapan Sistem Kriptografi ElGamal atas $\mathbb{Z}_p^*$**
**Dalam Pembuatan Tanda Tangan Digital**

Telah Disetujui dan Disahkan pada Tanggal 6 April 2011
Untuk Dipertahankan Didepan Panitia Penguji Skripsi
Program Studi Matematika
Jurusan Pendidikan Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Negeri Yogyakarta

Menyetujui,

Pembimbing

Dr. Agus Maman Abadi , M.Si
NIP.197008281995021001

## PERNYATAAN

Yang bertanda tangan di bawah ini :

Nama Mahasiswa : Rininda Ulfa Arizka

NIM : 07305141010

Jurusan/ Prodi : Pendidikan Matematika/ Matematika

Fakultas : MIPA

Judul TAS : Penerapan Sistem Kriptografi ElGamal atas $\mathbb{Z}_p{}^*$ Dalam Pembuatan Tanda Tangan Digital

Menyatakan bahwa skripsi ini adalah hasil pekerjaan saya sendiri dan sepanjang pengetahuan saya, tidak berisi materi yang dipublikasikan atau ditulis oleh orang lain atau telah digunakan sebagai persyaratan penyelesaian studi di Perguruan Tinggi lain kecuali pada bagian-bagian tertentu yang saya ambil sebagai acuan.

Apabila ternyata terbukti pernyataan ini tidak benar, sepenuhnya menjadi tanggungjawab saya dan saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Yogyakarta, 1 April 2011

Yang Menyatakan

Rininda Ulfa Arizka

NIM. 07305141010

**PENGESAHAN**

**Skripsi**

**Penerapan Sistem Kriptografi ElGamal atas $\mathbb{Z}_p^*$ Dalam Pembuatan Tanda Tangan Digital**

Oleh :

Rininda Ulfa Arizka
07305141010

Telah Dipertahankan Di Depan Panitia Penguji Skripsi Program Studi Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Yogyakarta pada tanggal 15 April 2011 dan dinyatakan telah memenuhi syarat guna memperoleh gelar sarjana sains.

**DEWAN PENGUJI**

| Nama | Jabatan | Tanda Tangan | Tanggal |
|---|---|---|---|
| Dr. Agus Maman A | Ketua Penguji | …… | …… |
| Tuharto , M.Si | Sekretaris Penguji | …… | …… |
| Dr. Hartono | Penguji Utama | …… | …… |
| Dr. Sugiman | Anggota Penguji | ……. | …… |

Yogyakarta, 18 April 2011
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Negeri Yogyakarta
Dekan,

Dr. Ariswan
NIP. 195909141988031003

## MOTTO

*Tak semua yang dapat dihitung , diperhitungkan , dan tak semua yang diperhitungkan dapat dihitung*

***Albert Einstein***

*Sabar adalah tirai untuk menutupi, dan akal adalah pedang yang tajam. Karena itu simpanlah kelemahan dan perilaku Anda dengan kesabaran Anda, dan bunuhlah hawa nafsu Anda dengan akal Anda.*

***-Ali bin Abi Thalib***

*Jadikanlah sabar dan sholat sebagai penolongmu. Dan sesungguhnya yang demikian itu sungguh berat, kecuali bagi orang-orang yang khusu'*

***-Qs. Al Baqarah : 5***

*Barang siapa menempuh jalan untuk mendapatkan ilmu, Allah akan memudahkan baginya jalan menuju surga*

***(HR. Muslim)***

*Seluruh kesulitan dalam hidup ini adalah bagian dari suatu tatanan yang sempurna dan sifat yang paling pasti dari sistem tata surya ini.*

***-Pierre Simon de Laplace-***

*Sesungguhnya sesudah kesulitan itu ada kemudahan. Maka apabila kamu telah selesai (dari sesuatu urusan), kerjakanlah dengan sungguh-sungguh (urusan) yang lain. Dan hanya kepada Tuhanmulah hendaknya kamu berharap.*

***-QS. Al Insyirah : 6 – 8***

**PERSEMBAHAN**

*Alhamdullillahirobbil'alamin, karya ini penulis persembahkan kepada :*

1. *Ibu dan Bapak tersayang ,*
   *yang telah memberikan makna dalam hidup,*
   *Teriring ucapan terimakasih yang dalam . . . .*
   *Atas segala do'a, kasih sayang, pengertian, dukungan, dan kesabaran*
   *dari aku kecil hingga dewasa . . . .*

2. *Untuk kakakku, Harizal Rizki Ramadhian terimakasih atas do'a,*
   *kasih sayang, kesabaran dan seluruh pengertiannya*

3. *Untuk guru-guruku yang telah memberikan ilmu*
   *hingga sampai saat ini*

*Penulis mengucapkan terimakasih kepada :*

1. *Untuk sahabatku Mustofa ,Alin, Pupe dan Sinta*
   *terimakasih atas do'a, bantuan, dukungan,*
   *kebersamaan dan persahabatannya*

2. *Untuk mas Puguh Wahyu , yang telah membantu dalam*
   *proses penyelesaian skripsi ini*

3. *Untuk Isna, Arsluz, Eka, Nana, Erna, Bagus, Rizki, Anas, teman AK47*
   *serta rekan-rekan Matematika 2007 yang tak bisa disebutkan satu-*
   *persatu, terima kasih atas dukungannya*

# KATA PENGANTAR

Alhamdulillah, puji syukur penulis panjatkan ke hadirat Allah SWT atas segala limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan penulisan skripsi yang berjudul "*Penerapan Sistem Kriptografi ElGamal atas* $\mathbb{Z}_p^*$ *dalam Pembuatan Tanda Tangan Digital"* ini.

Penulis menyadari sepenuhnya bahwa dalam penulisan skripsi ini tidak terlepas dari dukungan, motivasi, kerjasama maupun bimbingan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terimakasih yang sebesar-besarnya kepada :

1. Bapak Dr. Ariswan, Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Yogyakarta yang telah memberikan kesempatan penulis dalam menyelesaikan studi.
2. Bapak Dr. Hartono, Ketua Jurusan Pendidikan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Yogyakarta yang telah memberikan kemudahan pengurusan administrasi sekaligus sebagai penguji utama skripsi saya.
3. Ibu Atmini Dhoruri, M.Si, Ketua Program Studi Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Yogyakarta yang telah memberi dukungan untuk kelancaran studi.

4. Bapak Dr. Agus Maman Abadi, M.Si, dosen pembimbing yang telah dengan sabar membimbing penulis dan selalu memberikan pengarahan dalam penulisan skripsi.
5. Ibu Caturiyati, M.Si, dosen penasehat akademik yang selalu memberikan motivasi kepada penulis.
6. Bapak Tuharto, M.Si dan Bapak Dr. Sugiman, M.Si, sekretaris penguji dan penguji pendamping skripsi saya, yang memberikan berbagai masukan yang membangun.
7. Seluruh dosen Jurusan Pendidikan Matematika FMIPA Universitas Negeri Yogyakarta yang telah memberikan ilmu kepada penulis.
8. Semua pihak yang telah membantu tersusunnya skripsi ini yang tidak dapat penulis sebutkan satu-persatu.

Penulis menyadari bahwa dalam skripsi ini masih banyak kekurangan . Oleh karena itu penulis mengharapkan kritik dan saran yang membangun untuk menyempurnakan skripsi ini. Akhir kata, penulis berharap semoga skripsi ini dapat memberikan sesuatu yang bermanfaat bagi semua pihak yang membacanya.

Yogyakarta, Maret 2011

Penulis

# DAFTAR ISI

## DAFTAR GAMBAR

# DAFTAR TABEL

## DAFTAR LAMPIRAN

## DAFTAR SIMBOL

| | |
|---|---|
| $x \in X$ | : $x$ anggota $X$ |
| $a \mid b$ | : $a$ membagi b |
| $\lfloor a \rfloor$ | : bilangan bulat terbesar yang lebih kecil atau sama dengan $a$ |
| $gcd(a,b)$ | : Faktor Persekutuan Terbesar dari $a$ dan $b$ |
| ■ | : tanda berakhirnya suatu bukti |
| $\phi(m)$ | : banyaknya elemen dari himpunan residu sederhana modulo $m$ |
| $\mathbb{Z}$ | : himpunan semua bilangan bulat |
| $\mathbb{C}$ | : himpunan semua bilangan cacah |
| $\mathbb{R}$ | : himpunan semua bilangan real |
| $\mathbb{Q}$ | : himpunan semua bilangan rasional |
| $\mathbb{Z}_n$ | : himpunan semua bilangan bulat modulo $n$ |
| $\mathbb{Z}_p^*$ | : himpunan kelas bilangan bulat modulo $p$ yang saling prima dengan $p$ |
| $P$ | : himpunan semua *plaintext*. |
| $C$ | : himpunan semua *chipertext*. |

# DAFTAR ALGORITMA

**Penerapan Sistem Kriptografi ElGamal atas $\mathbb{Z}_p^*$ Dalam Pembuatan Tanda Tangan Digital**

**Oleh :**

**Rininda Ulfa Arizka**
**07305141010**

## ABSTRAK

Tanda tangan digital dapat digunakan untuk melakukan pembuktian secara matematis bahwa data tidak mengalami modifikasi secara ilegal, sehingga bisa digunakan sebagai salah satu solusi untuk melakukan verifikasi data. Tujuan dari penulisan skripsi ini adalah untuk mengetahui proses pembuatan tanda tangan digital menggunakan sistem kriptografi ElGamal atas $\mathbb{Z}_p^*$.

Proses pembuatan tanda tangan digital diawali dengan pembuatan kunci publik dan kunci privat. Pada proses pembentukan kunci dipilih $g$,dan $p$, dan dipilih bilangan rahasia $s \in \{1,2,3,\dots,p-1\}$. Kemudian dicari $v = g^s mod\ p$. Kunci publik dikirim kepada pengirim pesan. Proses selanjutnya adalah perhitungan nilai *hash* dari suatu pesan. Lalu memilih bilangan $e$ ,dengan $gcd(e,\ p\text{-}1)=1$ dan menghitung $R = g^e mod\ p$ serta $T = (MD - sR)e^{-1} mod\ (p-1)$. Diperoleh tanda tangan ($R$,$T$) yang kemudian dibubuhkan pada dokumen dan dikirimkan. Setelah menerima dokumen, maka penerima akan memverifikasi tanda tangan, dengan mencari nilai *MD* dahulu, lalu mengecek bahwa $1 \leq R \leq p\text{-}1$, jika terpenuhi lanjut menghitung $v^R R^T \bmod p$, selanjutnya diperiksa bahwa $v^R R^T \equiv g^{MD} \bmod p$.

Proses pembuatan kunci menghasilkan kunci publik ($p$, $g$, $v$ ) dan kunci privat $s$. Kunci publik akan dikirimkan kepada penerima pesan untuk memverifikasi tanda tangan. Pada proses perhitungan nilai *hash* akan dihasilkan *message digest*, yang akan digunakan dalam pembuatan tanda tangan. Proses penandatanganan dihasilkan sepasang tanda tangan ($R$, $T$). Tanda tangan dan dokumen dikirimkan kepada penerima. Selanjutnya, pada proses verifikasi, penerima akan mengecek apakah tanda tangan tersebut cocok atau tidak dengan menggunakan kunci publik dan menghitung nilai *hash* dokumen yang ia terima.

**Kata Kunci :** tanda tangan digital, ElGamal, fungsi *hash*, kriptografi, kunci publik, tanda tangan digital ElGamal.

# BAB I

# PENDAHULUAN

## A. Latar Belakang

Dewasa ini, perkembangan ilmu dan teknologi telah mempengaruhi segala aspek kehidupan, tak terkecuali aspek komunikasi, seperti dalam pengiriman pesan. Semakin berkembangnya teknologi, pengiriman suatu pesan juga menjadi kurang aman. Tidak menutup kemungkinan saat proses pengiriman pesan tersebut ada pihak ketiga yang ingin merubah dari pesan tersebut. Salah satu cara untuk mempertahankan kerahasiaan dari pesan tersebut, maka pesan yang akan dikirimkan disandikan menjadi kode-kode yang tidak dipahami, sehingga bila ada pihak ketiga yang ingin merubah akan kesulitan dalam menterjemahkan isi pesan yang sebenarnya. Namun, hanya dengan menyandikan pesan tersebut, tidak menutup kemungkinan pesan dirubah oleh pihak ke tiga. Untuk memperkuat kerahasiaan serta keaslian dari pesan tersebut, maka berkembanglah tanda tangan digital. Penerima pesan akan percaya bahwa pesan yang dikirimkan masih otentik, karena telah dibubuhkan tanda tangan pada pesan tersebut. Selanjutnya, untuk mengatasi permasalahan di atas, dapat diselesaikan dengan kiptografi. Kriptografi tidak hanya menyediakan alat untuk keamanan pesan, tetapi juga sekumpulan teknik yang berguna (Rinaldi,2006 :2).

Kriptologi berasal dari bahasa Yunani, yang terdiri dari dua kata yaitu *cryptos* dan *graphein*. *Cryptos* berarti rahasia, dan *graphein* berarti tulisan. Sehingga menurut bahasa, kriptologi berarti tulisan rahasia. Sedangkan definisi

kriptografi adalah suatu ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi, integritas suatu data, serta otentifikasi data (Menezes,1996 : 4). Enkripsi adalah sebuah proses penyandian yang melakukan perubahan sebuah kode (pesan) dari yang bisa dimengerti (*plaintext*) menjadi sebuah kode yang tidak bisa dimengerti (*ciphertext*). Sedangkan proses kebalikannya untuk mengubah *ciphertext* menjadi *plaintext* disebut dekripsi. Proses enkripsi dan dekripsi memerlukan suatu mekanisme dan kunci tertentu.

Kriptografi dapat dibedakan menjadi kriptografi kunci simetri (*symmetric-key cryptography*) dan kriptografi kunci asimetri (*asymmetric-key cryptography*). Pada kriptografi kunci simetri, kunci untuk proses enkripsi sama dengan kunci pada proses dekripsi. Jadi dalam hal ini, pengirim dan penerima pesan sudah berbagi kunci sebelum saling bertukar pesan. Contoh algoritma kriptografi kunci simetri adalah DES (*Data Encryption Standard*), AES (*Advanced Encryption Standard*), dan lain sebagainya. Kelemahan dari sistem ini adalah pada pemakaian kunci yang sama, sehingga pengirim pesan harus mencari cara yang aman untuk menyampaikan kunci kepada penerima pesan. Pada kriptografi kunci asimetri ( kriptografi asimetri ), kunci yang digunakan pada proses enkripsi dan dekripsi adalah berbeda. Kunci untuk enkripsi bersifat tidak rahasia, bisa diketahui oleh publik, sering dinamakan sebagai kunci publik. Sedangkan kunci untuk dekripsi, bersifat rahasia, hanya diketahui oleh penerima pesan saja. Beberapa contoh sistem kriptografi asimetri yang sering digunakan pada saat ini adalah RSA dan ElGamal. Sistem kunci publik RSA, yang ditemukan oleh River, Shamir, dan

Adleman, dikembangkan berdasarkan teori bilangan. Keamanan sistem RSA terletak pada kesulitan pemfaktoran bilangan – bilangan besar. Tetapi hal ini juga menunjukan adanya kelemahan bahwa seiring dengan kemajuan teknologi komputasi, problem pemfaktoran bilangan besar juga akan mudah diatasi. Kelemahan lain dari sistem RSA adalah terlalu lambat pada proses enkripsi (pengkodean) data. Sistem kriptografi ElGamal, merupakan contoh sistem kriptografi logaritma diskret. Dalam hal ini kunci publik yang dibangun berdasarkan logaritma diskret tidak bisa diterobos (*unbreakable*). Sistem kriptografi ElGamal dikembangkan pertama kali oleh Taher Gamal pada tahun 1984. Sampai saat ini sistem kriptografi ini masih dipercaya sebagai metode untuk pengamanan pesan. Berdasarkan uraian diatas,dalam skripsi ini,akan dibahas tentang menjaga keotentikan suatu dokumen,yaitu dengan cara pembuatan tanda tangan digital dengan menggunakan sistem kriptografi ElGamal atas $\mathbb{Z}_p^*$ . Dimana $\mathbb{Z}_p^*$ = {1, 2, 3, 4, ..., $p$-1} adalah himpunan bilangan bulat modulo $p$ yang saling prima dengan $p$.

**B. Pembatasan Masalah**

Dalam penulisan skripsi ini, pembahasan sistem kriptografi ElGamal atas $\mathbb{Z}_p^*$ dan dibatasi pada konsep – konsep matematis yang melandasi sistem kriptografi ElGamal atas $\mathbb{Z}_p^*$ , serta proses penyandiannya pada proses pengiriman pesan, untuk proses perhitungannya digunakan suatu program aplikasi, yaitu *Matlab2009*, termasuk menguji bilangan prima yang digunakan, menghitung fungsi *hash*, dan sebagainya. Pada skripsi ini tidak membahas mengenai pengembangan sistem kriptografi yang lain atau aplikasinya pada

bidang yang lain dan skripsi ini tidak membahas mengenai kesulitan dan cara-cara untuk memecahkan mekanisme persandian.

**C. Rumusan Masalah**

Berdasarkan latar belakang masalah di atas, maka dirumuskan pokok permasalahan yang akan menjadi kajian dari skripsi ini, yaitu "Bagaimana proses pembuatan tanda tangan digital ElGamal atas $\mathbb{Z}_p^*$ ?".

**D. Tujuan Penelitian**

Tujuan dilaksanakan penelitian ini adalah menjelaskan konsep – konsep matematis tentang proses pembuatan tanda tangan digital dengan menggunakan sistem kriptografi ElGamal atas $\mathbb{Z}_p^*$ .

**E. Manfaat Penulisan**

Dari hasil penelitian ini diharapkan dapat memberi informasi tentang pengembangan sistem kriptografi ElGamal atas $\mathbb{Z}_p^*$ dalam pembuatan tanda tangan digital dengan berlandaskan teori bilangan, serta struktur aljabar. Pemahaman bagi pembaca dan pihak – pihak yang berkecimpung dalam kriptografi.

# BAB II

# DASAR TEORI

Pada bab ini akan dibahas tentang konsep dasar yang berhubungan dengan kriptografi. Adapun yang dibahas adalah tentang fungsi, bilangan bulat, bilangan prima, kekongruenan, dan perkongruenan linier. Sistem kriptografi Elgamal berdasar pada grup $\mathbb{Z}_p^*$, oleh karena itu penting bahwa pada bab ini dibahas mengenai grup serta struktur aljabar lainnya, serta gelanggang dan sifat-sifatnya.

## A. Fungsi

Berikut akan diberikan definisi tentang fungsi injektif, fungsi surjektif, fungsi bijektif, dan fungsi invers.

### 1. Fungsi Injektif (Fungsi satu-satu )

**Definisi 2.1.(Sukirman, 2005 :9)**

Pemetaan $f : S \rightarrow T$ disebut injektif jika dan hanya jika $\forall\ x \in f(S)$ , $f^{-1}(x)$ adalah himpunan tunggal yang hanya memuat satu elemen.

Berdasarkan definisi tersebut dapat dikatakan bahwa setiap elemen dari daerah hasil mempunyai prapeta tepat satu elemen dari daerah asal, yang berarti setiap dua elemen yang berbeda dalam daerah asal mempunyai peta yang berbeda pula dalam daerah kawan, dan dapat dituliskan sebagai berikut:

Fungsi $f : S \rightarrow T$ injektif $\Leftrightarrow \forall\, x,\ y \in S,\ x \neq y \Rightarrow f(x) \neq f(y)$

Kontraposisinya yaitu :

Fungsi $f : S \rightarrow T$ injektif $\Leftrightarrow \forall\, x,\ y \in S,\ f(x) = f(y) \Rightarrow x = y.$

**Contoh 2.1**

Misalkan U adalah himpunan bilangan asli, dan pemetaannya $\theta$ : U → U didefinisikan oleh $\theta(x)= 2x + 1$, $\forall x \in$ U. Maka pemetaan ini suatu pemetaan injektif, sebab jika $x, y \in$ U sedemikian sehingga $f(x) = f(y)$, yaitu $2x+1 = 2y+1$, maka $x = y$

**2. Fungsi Surjektif**

**Definisi 2.2. (Sukirman, 2002:11)**

Fungsi $f : S \rightarrow T$ disebut fungsi surjektif (onto) jika dan hanya jika setiap elemen dari daerah kawan merupakan peta dari suatu elemen dari daerah asal. Sehingga dapat ditulis secara simbolik sebagai berikut :

Fungsi $f : S \rightarrow T$ dikatakan surjektif $\Leftrightarrow \forall y \in T, \exists s \in S \ni f(s) = t$

**Contoh 2.2** :

Misalkan $\mathbb{Z}$ adalah himpunan semua bilangan bulat, dan $\mathbb{C}$ adalah himpunan bilangan cacah. Pemetaan f : $\mathbb{Z}$ → $\mathbb{C}$ didefinisikan oleh $f(x) = |x|$, $\forall x \in \mathbb{Z}$.

Ambil $c \in \mathbb{C}$, lalu ditentukan $z \in \mathbb{Z}$ , sedemikian sehingga

$f(x) = |z| = c$

$\Rightarrow z = c$

$\Rightarrow f(\mathbb{Z}) = \mathbb{C}$

Dengan demikian, setiap elemen dari $\mathbb{C}$ pasti mendapatkan pasangan dengan elemen dari $\mathbb{Z}$.

**3. Fungsi Bijektif (Fungsi Korespondensi Satu-satu )**

**Definisi 2.3. (Sukirman, 2005:10)**

Fungsi yang sekaligus injektif dan surjektif disebut fungsi bijektif (satu-satu dan onto) atau korespondensi satu – satu.

**Contoh 2.3 (Sukirman, 2005 :10)**

Misalkan $\mathbb{R}$ adalah himpunan semua bilangan real. Fungsi $g : \mathbb{R} \rightarrow \mathbb{R}$ didefinisikan oleh $g(x) = 4x + 3, \forall x \in \mathbb{R}$.

Pemetaan ini injektif sebab jika $a, b \in \mathbb{R}$ sedemikian hingga $g(a) = g(b)$, yaitu $4a + 3 = 4b + 3$, maka $a = b$.

Pemetaan ini surjektif sebab jika $d \in \mathbb{R}$, maka terdapat $c \in \mathbb{R}$ dengan $c = \frac{d-3}{4}$ sedemikian hingga $g(c) = g(\frac{d-3}{4}) = 4(\frac{d-3}{4}) + 8 = d$

Karena pemetaan ini injektif dan surjektif maka dapat disimpulkan bahwa pemetaan ini adalah pemetaan bijektif.

**4. Fungsi Invers**

**Definisi 2.4. (Rinaldi, 2006 :30)**

Jika $f$ adalah fungsi bijektif dari $S$ ke $T$, maka dapat ditemukan invers dari $f$, dilambangkan dengan $f^{-1,}$ yang memetakan $T$ ke $S$ sebagai berikut :

Misalkan $s$ adalah anggota himpunan dari $S$ dan $t$ adalah anggota himpunan dari $T$. Dengan demikian , $f^{-1}(t) = s$ apabila $f(s) = t$

**Contoh 2.4**

1. Misalkan $\mathbb{R}$ adalah himpunan semua bilangan real. Fungsi $f : \mathbb{R} \rightarrow \mathbb{R}$ didefinisikan oleh $f(x) = 4x + 3, \quad \forall x \in \mathbb{R}$. Pada contoh 2.3 telah

dibuktikan bahwa *f* merupakan korespondensi satu – satu. Akan dicari fungsi invers dari fungsi *f*.

$y = 4x + 3$

$\Rightarrow \quad -4x = -y + 3$

$\Rightarrow \quad x = \frac{y-3}{4}$

Jadi fungsi invers dari *f* adalah $f^{-1}(x) = \frac{x-3}{4}$.

2. Diberikan suatu fungsi kuadrat $f(x) = x^2+1$. Fungsi kuadrat ini bukan merupakan fungsi invers, karena fungsi ini bukan merupakan fungsi bijektif. Fungsi ini merupakan fungsi invers jika $x = [0,\infty]$.

## B. Teori Bilangan

Teori bilangan merupakan suatu teori yang mendasar dalam mempelajari kriptografi, khususnya pada kriptografi dengan kunci publik (kriptografi kunci asimetri).Jenis bilangan yang dimaksudkan dalam hal ini adalah bilangan bulat (*integer*). Bilangan bulat adalah bilangan yang tidak mempunyai pecahan desimal. Bilangan – bilangan bulat dinyatakan dengan huruf – huruf latin kecil *a, b, c, ..., m, n,* dan sebagainya yang dapat bernilai positif, nol atau negatif. Himpunan semua bilangan bulat yang dinotasikan dengan $\mathbb{Z}$ adalah himpunan {..., -3, - 2, -1, 0, 1, 2, 3, ...}. Pada himpunan bilangan bulat, berlaku sifat-sifat asosiatif, komutatif, dan distributif dalam operasi penjumlahan dan perkalian.

### 1. Keterbagian

**Definisi 2.5 ( Buchmann, 2000 :13)**

Bilangan bulat *a* dikatakan membagi (habis) $n \neq 0$ jika terdapat sebuah bilangan bulat *b*, dengan $n = a\ b$. Jika *a* membagi (habis) *n*, maka *a* dikatakan

pembagi dari $n$. Dan $n$ dikatakan kelipatan dari $a$, dan dapat dituliskan $a|n$. Jika $a$ tidak membagi (habis) $n$, dapat dituliskan $a \nmid n$.

Bilangan bulat $b$ pada Definisi 2.5 bersifat tunggal, sebab apabila ada bilangan bulat $m$ selain $b$ sedemikian sehingga $n = ab$ dan $n = am$, maka $ab = am$, sehingga $b = m$.

Jika $a = 0$ dan $n \neq 0$ , maka tidak ada bilangan $b$ yang memenuhi $n = ab$. Akan tetapi jika $a = 0$ dan $n = 0$, maka terdapat tak hingga bilangan bulat $b$ yang memenuhi $n = ab$.

**Contoh 2.5**

Diberikan $n = 48$, maka

$8|48$, karena ada bilangan bulat yaitu 6, sedemikian sehingga $8.6 = 48$.

$7 \nmid 48$, karena tidak ada bilangan bulat $m$, sedemikian sehingga $7.m = 48$.

**Teorema 2.1 (Buchmann, 2000 : 14)**

1. Jika $a \mid b$, dan $b \mid c$ , maka $a \mid c$.
2. Jika $a \mid b$, maka $ac \mid bc$ , untuk setiap bilangan bulat $c$.
3. Jika $c \mid a$ dan $c \mid b$ , maka $c \mid da + eb$, untuk setiap bilangan bulat $d$ dan $e$ .
4. Jika $a \mid b$, dan $b \neq 0$, maka $|a| \leq |b|$.
5. Jika $a \mid b$ dan $b \mid a$ , maka $|a| = |b|$.

**Bukti :**

1. Jika $a|b$ dan $b|c$ , maka terdapat $f,\ g \in \mathbb{Z}$ sedemikian hingga $b = a.f$ dan $c = b.g$ . Diperoleh $c = b.g = (a.f).g = a.(f.g)$. Karena $f, g \in \mathbb{Z}$ , maka $a|c$.

2. Jika $a \mid b$, maka terdapat $d \in \mathbb{Z}$ sedemikian sehingga, $b = a.d$. sebagai akibatnya , untuk sebarang $c \in \mathbb{Z}$, akan diperoleh $b.c = (a.d).c = (d.a).c = d.(a.c)$. Terbukti bahwa $a.c \mid b.c$ , untuk setiap $c \in \mathbb{Z}$.
3. Jika $c \mid a$ dan $c \mid b$, maka terdapat $f, g \in \mathbb{Z}$. Diperoleh $a = f.c$ dan $b = g.c$. Ini mengakibatkan untuk sebarang $d, e \in \mathbb{Z}$, $d.a + e.b = d.(f.c) + e.(g.c) = (d.f).c + (e.g).c = c.(d.f + e.g)$ , dengan kata lain $c \mid (d.a + e.b)$.
4. Jika $a \mid b$ dan $b \neq 0$ , dan terdapat $f \neq 0$, dengan $b = a.f$. Ini mengakibatkan $|b| = |a.f| \geq |a|$ , dengan kata lain $|a| \leq |b|$.
5. Diketahui bahwa $a \mid b$, dan $b \mid a$. Jika $a = 0$ , $b = 0$ , dan sebaliknya, jika $a \neq 0$, dan $b \neq 0$ , dengan menggunakan bukti (4) diperoleh bahwa $|a| \leq |b|$ dan $|a| \geq |b|$. Maka $|a| = |b|$. ■

**2. Algoritma Pembagian pada Bilangan Bulat**

**Definisi 2.6. (Buchmann, 2000 : 2)**

Untuk setiap bilangan real $\alpha \in \mathbb{R}$ didefinisikan $\lfloor \alpha \rfloor = \max \{ b \in \mathbb{Z} : b \leq \alpha \}$.

Oleh sebab itu, $\lfloor \alpha \rfloor$ merupakan bilangan bulat terbesar yang lebih kecil atau sama dengan $\alpha$.

**Contoh 2.6.( Buchmann, 2000 : 2)**

1. $\lfloor 3,43 \rfloor = 3$
2. $\lfloor -3,43 \rfloor = -4$
3. $\lfloor 3 \rfloor = 3$

**Teorema 2.2. (Buchmann, 2000 : 3)**

Jika $a$ dan $b$ adalah bilangan bulat, $b > 0$, maka terdapat tunggal bilangan bulat $q$ dan $r$ , sehingga dengan demikian $a = q.b + r$ dengan $0 \leq r < b$, yakni $q = \lfloor a/b \rfloor$ dan $r = a - b.q$ .

**Bukti :**

Diambil bilangan bulat $a$ dan $b$ dengan $b > 0$, akan ditunjukkan bahwa terdapat $q = \lfloor a/b \rfloor \in \mathbb{Z}$ dan $r \in \mathbb{Z}$ sedemikian sehingga $a = bq + r$ dengan $0 \leq r < b$. Diketahui bahwa $a$, $b \in \mathbb{Z}$ dan $b > 0$, menggunakan Definisi 2.2. diperoleh bilangan bulat $q = \lfloor a/b \rfloor$, sehingga diperoleh $a \geq bq$. Akibatnya terdapat $r \in \mathbb{Z}$, $r \geq 0$ sehingga $a = bq + r$. Jika $b$ pembagi dari $a$, maka $a = bq$ sehingga diperoleh $r = 0$. Jika $b$ bukan pembagi dari $a$, maka $a = qb + r$ dengan hasil bagi $q = \lfloor a/b \rfloor \in \mathbb{Z}$, dan $r \in \mathbb{Z}$ adalah sisa $a$ dibagi $b$. Jika diambil $r = b$, maka $a = b.q+b = b(q+1)$ sehingga $q = \lfloor a/b - 1 \rfloor$, akibatnya terjadi kontradiksi dengan yang diketahui yaitu $q = \lfloor a/b \rfloor$. Selanjutnya, dari hasil terakhir dan karena $b > 0$. Maka $0 \leq r < b$. Misalkan terdapat $q_1$, $q_2$, $r_1$, $r_2 \in \mathbb{Z}$ sedemikian hingga $a = q_1b + r_1$ dan $a = q_2b + r_2$. Akibatnya diperoleh $(q_1b + r_1) - (q_2b + r_2) = 0$ atau $b(q_1-q_2) + (r_1-r_2) = 0$. Karena $q_1 = \lfloor a/b \rfloor$ dan $q_2 = \lfloor a/b \rfloor$, maka $q_1 = q_2$, sehingga $q_1 - q_2 = 0$. Akibatnya $r_1-r_2 = 0$ atau $r_1 = r_2$. Disimpulkan bahwa $q$ dan $r$ tunggal. ■

**Contoh 2.7 :**

Diberikan suatu bilangan bulat 16 dan 56. Menggunakan Definisi 2.2 diperoleh bilangan bulat $\lfloor 56/16 \rfloor = \lfloor 3,5 \rfloor = 3$ . Menggunakan teorema 2.2 terdapat bilangan tunggal q dan r sedemikian sehingga $56 = 16.q + r$ , dengan

$0 \leq r < 16$ , yaitu q = 3, dan r = 8, dapat dilihat bahwa 56 = 16.3 + 8, dan $0 \leq 8 < 16$.

**4. Faktor Persekutuan Terbesar**

Berikut ini akan dijelaskan pengertian dan sifat-sifat dari suatu bilangan yang disebut dengan faktor persekutuan terbesar.

**Definisi 2.7 (Sukirman, 2006 : 38)**

Jika *a* dan *b* adalah bilangan-bilangan bulat, maka bilangan bulat *d* disebut faktor persekutuan dari *a* dan *b* jika $d|a$ dan $d|b$ .

**Contoh 2.8:**

Diberikan suatu bilangan 40 dan 50 $\in \mathbb{Z}$ . Faktor persekutuan dari 40 dan 50 adalah 10, karena 10 membagi 40 dan 50.

**Definisi 2.8 (Sukirman, 2006 :39)**

Jika *a* dan *b* bilangan – bilangan bulat yang sekurang – kurangnya satu di antara tidak sama dengan nol, maka faktor persekutuan terbesar (FPB) atau *greatest common divisor* (*gcd*) dari *a* dan *b* diberi simbol *gcd*(*a*, *b*) adalah suatu bilangan bulat positif, misalnya *d*, yang memenuhi :

(i) $d \mid a$ dan $d \mid b$, serta

(ii) Jika $e \mid a$ dan $e \mid b$, maka $e \leq d$.

Berdasarkan definisi tersebut dapat disimpulkan bahwa jika *gcd*(*a*, *b*) = *d*, maka $d \geq 1$. Apabila ada faktor persekutuan lain, misalkan *e* maka $e \leq d$.

**Contoh 2.9.**

Diberikan bilangan bulat 25 dan 40, maka :

Faktor – faktor persekutuan dari 25 dan 40 adalah 1 dan 5, sehingga $gcd(25, 40) = 5$.

**Teorema 2.3 (Sukirman, 2006 : 40)**

Jika $gcd(a, b) = d$, maka $gcd(a : d, b : d) = 1$

**Bukti :**

Jika $gcd(a, b) = d$ maka dapat disimpulkan bahwa $d$ adalah bilangan bulat positif terbesar sedemikian hingga $d|a$ dan $d|b$. Karena $d|a$ maka terdapat bilangan bulat $r$ sedemikian hingga $a = rd$ akibatnya $a:d = r$. Demikian juga untuk $d|b$ maka terdapat $s$ anggota bilangan bulat sedemikian hingga $b = sd$ akibatnya $b:d = s$. Karena $d$ merupakan faktor persekutuan terbesar dari $a$ dan $b$, maka bilangan bulat $r$ dan $s$ saling prima sehingga $gcd(r, s) = 1$. Jadi dapat disimpulkan bahwa $gcd(a:d, b:d) = gcd(r, s) = 1$.

Apabila $a$ dan $b$ dua bilangan bulat positif dengan $gcd(a, b) = 1$, maka dikatakan bahwa $a$ dan $b$ saling prima atau $a$ prima relalif terhadap $b$. ■

**5. Algoritma Euclide**

Berikut ini diberikan suatu algoritma yang dapat digunakan untuk menghitung nilai pembagi persekutuan terbesar dari dua buah bilangan bulat dengan sangat efisien. Algoritma ini didasarkan pada teorema di bawah ini.

**Teorema 2.4 (Buchmann, 2000 : 12)**

1. Jika $b = 0$ , maka $gcd(a, b) = a$
2. Jika $b \neq 0$ , maka $gcd(a, b) = gcd(b, a \bmod b)$

**Bukti :**

1. Jelas bahwa $gcd(a, b) = gcd(a, 0) = a$.
2. Misalkan $d = gcd(a, b)$ dan $r = a \bmod b$ . Menurut Teorema 2.2 , terdapat $q \in \mathbb{Z}$ dengan $a = q.b + r$. Karena $r = a - b.q$ , maka $d \mid r$ . Selanjutnya akan ditunjukkan bahwa $d = gcd(b, r)$. Diambil sebarang bilangan bulat $t$ sedemikian sehingga $t|b$ dan $t|r$, yang terdapat $n.m \in \mathbb{Z}$ sehingga $b = n.t$ dan $r = m.t$ , diperoleh bahwa $a = q.b + r = q(n.t) + (m.t) = t(n.q + m)$ atau $t|a$. Diketahui bahwa $d=gcd(a, b)$, karena $t|a$ dan $t|b$ maka $t \leq d$ dan $t|d$. Terbukti bahwa $d=gcd(a, b)=gcd(b, a \bmod b)$.

   Misalkan diberikan bialngan bulat positif $r_0$ dan $r_1$ dengan $r_0 \geq r_1$. Selanjutnya dihitung algoritma pembagian sebagai berikut.

$$
\begin{aligned}
r_o &= q_1.a + r_2, \quad 0 < r_2 < b \\
r_1 &= q_2.r_2 + r_3, \quad 0 < r_3 < r_2 \\
r_2 &= q_3.r_3 + r_4, \quad 0 < r_4 < r_3 \\
r_3 &= q_4.r_4 + r_5, \quad 0 < r_5 < r_4 \\
r_4 &= q_5.r_5 + r_6, \quad 0 < r_5 < r_4 \\
&\ldots. \\
r_{k-2} &= q_{k-1}.r_{k-1} + r_k, \qquad 0 < r_k < r_{k-1} \\
r_{k-1} &= q_k.r_k + 0.
\end{aligned}
$$

   Dengan menggunakan Teorema 2.4 dapat ditunjukkan bahwa $gcd(r_0 , r_1) = gcd(r_1, r_2) = gcd(r_2, r_3) = ....= gcd(r_{k-2}, r_{k-1}) = gcd(r_{k-1}, r_k) = gcd(r_k, 0) = r_k$ . ■

**Contoh 2.10 :**

Akan dihitung nilai *gcd*(120, 35) dengan menggunakan algoritma Euclide diperoleh:

1. *gcd*(120, 35) = *gcd*(35, 120 mod 35) = *gcd*(35, 15)
2. *gcd*(35, 15) = *gcd*(15, 35 mod 15) = *gcd*(15, 5)
3. *gcd*(15, 5) = *gcd*(5, 15 mod 5) = *gcd*(5, 0)
4. *gcd*(5, 0)=5

jadi gcd (120, 35) = gcd (35, 15) = gcd(15, 5) =gcd(5, 0) = 5.

**Tabel 2.1. Perhitungan gcd(120, 35) menggunakan algoritma Euclid**

| $k$ | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| $a_k$ | 120 | 35 | 15 | 5 | 0 |
| $q_k$ | | 3 | 2 | 3 | |

## 5. Kekongruenan

**Definisi 2.9 (Sukirman, 2006 :87)**

Jika *m* suatu bilangan bulat positif, maka *a* kongruen dengan *b* modulo *m* (ditulis $a \equiv b \pmod{m}$) bila *m* membagi (*a*-*b*). Jika *m* tidak membagi (*a*-*b*) maka dikatakan *a* tidak kongruen dengan *b* modulo *m* ($a \not\equiv b \pmod{m}$).

**Contoh 2.11 (Sukirman, 2006 : 87)**

$25 \equiv 1 \pmod 4$, sebab (25-1)=24 terbagi habis oleh 4.

$31 \not\equiv 5 \pmod 6$, sebab (31-5) = 26 tidak terbagi habis oleh 6.

**Teorema 2.5 (Sukirman , 2006 :88)**

$a \equiv b \pmod m$ bila dan hanya bila ada bilangan bulat *k* sehingga $a = mk + b$.

**Contoh 2.12**

$43 \equiv 1 \pmod 7$ sama artinya dengan $43 = 7.6 + 1$.

**Definisi 2.10 (Sukirman, 2006: 88)**

Jika $a \equiv r \pmod{m}$ dengan $0 \leq r < m$, maka $r$ disebut residu terkecil dari $a$ modulo $m$. Untuk kekongruenan modulo $m$ ini, $\{0, 1, 2, 3, ..., (m-1)\}$ disebut himpunan residu terkecil modulo $m$.

**Contoh 2.13**

Residu terkecil dari 23 modulo 4 adalah 3, karena 23= 5.4 +3

**Contoh 2.14**

Himpunan residu terkecil dari modulo 8 adalah {0, 1, 2, 3, 4, 5, 6, 7}.

**Teorema 2.5 (Sukirman, 2006 :89)**

$a \equiv b \pmod{m}$ bila dan hanya bila $a$ dan $b$ memiliki sisa yang sama jika dibagi dengan $m$.

**Teorema 2.6 (Sukirman, 2006 :92)**

Jika $a \equiv b \pmod{m}$ dan $c \equiv d \pmod{m}$ maka $a+c \equiv b+d \pmod{m}$.

**Bukti :**

$a \equiv b \pmod{m}$ yang berarti bahwa $a = ms + b$, untuk suatu bilangan bulat $s$.

$c \equiv d \pmod{m}$ yang berarti bahwa $c = mt + d$, untuk suatu bilangan bulat $t$.

Dari dua persamaan di atas, akan memberikan :

$a + c = (ms+b) + (mt + d)$

$a + c = m(s+t)+(b+d)$

$(a + c) – (b+d) = m( s+t)$

Hal ini berarti bahwa $a + c \equiv (b + c) \pmod{m}$. ■

**Teorema 2.7 (Sukirman, 2006 : 92)**

Jika $a \equiv b \pmod{m}$ dan $c \equiv d \pmod{m}$ maka $ax + cy \equiv bx + dy \pmod{m}$, untuk setiap bilangan bulat $x$ dan $y$.

**Bukti :**

$a \equiv b \pmod{m}$ yang berarti bahwa $a = ms + b$, untuk suatu bilangan bulat $s$.

$c \equiv d \pmod{m}$ yang berarti bahwa $c = mt + d$, untuk suatu bilangan bulat $t$.

Jika kedua ruas persamaan pertama dikalikan $x$ dan kedua ruas persamaan kedua dikalikan dengan $y$, maka diperoleh :

$ax = msx + bx$ dan

$cy = mty + dy$

selanjutnya menjumlahkan kedua ruas persamaan ini diperoleh :

$ax+cy = (msx+bx) + ( mty + dy )$

$ax+cy = m(sx+ty) +( bx+dy)$

Berdasarkan persamaan yang terakhir berarti bahwa $ax+cy \equiv (bx+dy) \pmod{m}$. ■

**Teorema 2.8 (Sukirman, 2006 :93)**

Jika $ac \equiv bc \pmod{m}$ dengan $gcd(c, m) = 1$, maka $a \equiv b \pmod{m}$

**Bukti :**

$ac \equiv bc \pmod{m}$ yang berarti bahwa $m \mid ( ac-bc)$ atau $m \mid c\ (a-b)$. Diketahui $gcd\ (c, m)=1$, maka $m|(a-b)$ berarti $a \equiv b \pmod{m}$. ■

**Contoh 2.15**

Tentukanlah bilangan – bilangan bulat $y$ yang memenuhi perkongruenan $3y \equiv 1 \pmod{7}$.

**Jawab**

Misalkan $1 \equiv 15 \pmod 7$, maka kita dapat mengganti 1 pada perkongruenan tersebut dengan 15, sehingga diperoleh $3y \equiv 15 \pmod 7$ dan $gcd(3, 7)=1$, selanjutnya membagi 3 pada ruas-ruas perkongruenan tersebut, sehingga diperoleh $y \equiv 5 \pmod 7$. Berarti bahwa $y= 5 +7k$, untuk setiap bilangan bulat $k$.

**6. Perkongruenan Linier**

**Definisi 2.11**

Perkongruenan linier adalah perkongruenan yang variabelnya berpangkat paling tinggi satu. Bentuk umum dari perkongruenan linier adalah $ax \equiv b \pmod m$ dengan $a \not\equiv 0 \pmod m$.

**Teorema 2.9 (Sukirman, 2006 : 108 )**

Jika $gcd\ (a, m) \nmid b$ maka perkongruenan linier $ax \equiv b \pmod m$ tidak memiliki solusi.

**Teorema 2.10 (Sukirman, 2006 : 108 )**

Jika $gcd(a, m) = 1$, maka perkongruenan linier $ax \equiv b \pmod m$ memiliki tepat satu solusi.

**Contoh 2.16 (Sukirman, 2006 : 110)**

Carilah $2^{-1} \pmod{13}$

**Jawab:**

Untuk mencari $2^{-1} \pmod{13}$, kita perlu menyelesaikan perkongruenan $2x \equiv 1 \pmod{13}$.

$$2x \equiv 1 \pmod{13}$$

$2x \equiv 14 (\text{mod } 13)$

$x \equiv 7 (\text{mod } 13)$

Jadi $2^{-1} (\text{mod } 13)$ adalah 7.

**Teorema 2.11 (Sukirman, 2006 :137)**

Jika *gcd*(*a, m*) = 1, maka residu-residu terkecil modulo *m* dari barisan : *a, 2a, 3a, ...,* (*m*-1)*a* adalah suatu permutasi dari 1, 2, 3, ..., (*m*-1).

**Teorema 2.12. Teorema Fermat (Sukirman, 2006 : 138)**

Jika p suatu bilangan prima dan (*a, p*) = 1, maka $a^{p-1} \equiv 1 (\text{mod } p)$.

**Bukti :**

Ambil sembarang bilangan prima *p* dan bilangan bulat *a*, sedemikian *gcd* (*a, p*)=1, maka meurut Teorema 2.10, residu-residu terkecil mod *p* dari *a, 2a, 3a, ...,* (*p*-1)*a* adalah suatu permutasi dari 1, 2, 3, ..., (*p*-1), sehingga hasil kali- hasil kalinya akan kongruen mod *p* juga, yaitu :

$a.2a.3a. \ldots (p-1)a \equiv 1.2.3....(p-1) (\text{mod } p)$

$a^{p-1}(1.2.3...(p-1)) \equiv (p-1)! (\text{mod } p)$

$a^{p-1}(p-1)! \equiv (p-1)! (\text{mod } p)$

Karena *gcd*(*p,* (*p-1*)!) = 1, maka $a^{p-1} \equiv 1 (\text{mod } p)$ ■

**7. Fungsi Euler**

**Definisi 2.12 (Sukirman, 2006 : 185)**

Sistem residu sederhana modulo m adalah himpunan semua bilangan bulat positif $r_i$ yang memenuhi *gcd*($r_i$, *m*) = 1 dengan $r_i \not\equiv r_j (\text{mod } m)$ untuk $i \neq j$.

**Definisi 2.13. Fungsi Euler (Sukirman, 2006 : 186)**

Misalkan *m* suatu bilangan bulat positif, maka $\phi(m)$ adalah banyaknya elemen dari himpunan residu sederhana modulo *m*.

**Contoh 2.17.**

Himpunan {1, 3, 5, 7} adalah himpunan semua residu sederhana modulo 8, sehingga $\phi(8) = 4$.

Apabila *p* suatu bilangan prima, maka setiap bilangan bulat positif yang kurang dari *p* selalu saling prima terhadap *p*, sehingga $\phi(p) = p\text{-}1$.

**Teorema 2.12 (Sukirman, 2006 :188)**

Apabila p suatu bilangan prima dan k suatu bilangan bulat positif, maka

$$\phi(p^k) = p^{k-1}(p-1)$$

**Teorema 2.13 .Teorema Euler (Sukirman, 2006 : 198)**

Jika *m* suatu bilangan bulat positif, dan $gcd(a, m) = 1$, maka $a^{\phi(m)} \equiv 1 \pmod{p}$

**Teorema 2.14. Teorema Wilson (Sukirman, 2006 :147)**

Jika *p* suatu bilangan prima, maka $(p-1)! \equiv -1 \pmod{p}$

## 8. Akar Primitif

**Definisi 2.14 ( Sukirman, 2006 :212)**

Jika $gcd\ (a, m\ ) = 1$, dan order dari *a* modulo *m* adalah $\phi(m)$, maka *a* dinamakan akar primitif dari *m*.

Dengan kata lain, suatu bilangan bulat positif $m$ dikatakan memiliki akar primitif $a$, apabila $a^{\phi(m)} \equiv 1 \pmod{p}$, dan $a^k \neq 1 \pmod{m}$, untuk semua bilangan bulat positif $k < \phi(m)$ .

**Teorema 2.15. Teorema Lagrange (Sukirman, 2006 : 216)**

Jika $p$ suatu bilangan prima dan $f$ adalah suatu polinomial berderajat $n$, maka perkongruenan $f(x) \equiv 0 \pmod{p}$ mempunyai sebanyak-banyaknya $n$ solusi.

**Teorema 2.16 (Sukirman, 2006 :218 )**

Jika $p$ suatu bilangan prima dan $d \mid p-1$, maka perkongruenan $x^d - 1 \equiv 0 \pmod{d}$ mempunyai tepat $d$ solusi.

**Bukti :**

Menurut Teorema Fermat, yaitu jika $p$ prima dan $gcd(a, p) = 1$ maka $a^{p-1} \equiv 1 \pmod{p}$. Ini berarti perkongruenan $x^{p-1} - 1 \equiv 0 \pmod{p}$ mempunyai tepat $p-1$ solusi, yaitu :

$$1,\ 2,\ 3,\ ...,\ p-1$$

Misalkan bahwa $d \mid p-1$, maka

$$x^{p-1} - 1 = (x^d - 1)(x^{p-1-d} + x^{p-1-2d} + .... + 1)$$
$$= (x^d - 1)f(x)$$

Menurut Teorema 2.15 , $f(x) \equiv 0 \pmod{p}$ memiliki solusi paling banyak $(p-1-d)$ solusi. Misalkan $x = a$ suatu solusi dari akar $x^{p-1} - 1 \equiv 0 \pmod{p}$ yang bukan solusi dari $f(x) \equiv 0 \pmod{p}$, maka $a$ suatu solusi dari $x^d - 1 \equiv 0 \pmod{p}$. Sebelumnya diketahui bahwa $0 \equiv a^{p-1} - 1 \equiv (a^d - 1)\, f(a) \pmod{p}$. Karena $p$ prima dan $p \nmid f(a)$, maka $p \mid (a^d - 1)$. Jadi $x^d - 1 \equiv 0 \pmod{p}$ mempunyai

sekurang-kurangnya $p-1-(p-1-d) = d$ solusi. Menurut Teorema 2.15 $x^d-1 \equiv 0 \pmod{p}$ mempunyai sebanyak-banyaknya $d$ solusi. Jadi perkongruenan tersebut memiliki tepat $d$ solusi. ■

**9. Uji Bilangan Prima**

Dalam kriptografi ElGamal, dalam proses pembentukan kunci, enkripsi, serta dekripsi digunakan bilangan prima. Pada subbab ini akan dijelaskan tentang menguji suatu bilangan apakah bilangan tersebut bilangan prima atau tidak. Salah satu caranya adalah dengan menggunakan Teorema Lucas.

**Teorema 2.17**

Bila terdapat bilangan bulat $a$ sedemikian sehingga $a^{n-1} \equiv 1 \pmod{n}$ dan $a^{(n-1)/p} \neq 1 \pmod{n}$ untuk semua bilangan prima $p$ yang membagi $n-1$, maka $n$ adalah bilangan prima.

**Contoh 2.18**

Misalkan $n = 347$ dan $a = 2$ maka diperoleh $2^{346} \equiv 1 \pmod{347}$. Karena $n-1 = 2.173$ kemudian dilakukan perhitungan berikut :

$2^{346/2} = 2^{173} \equiv 346 \pmod{347}$

$2^{346/173} = 2^2 \equiv 4 \pmod{347}$

Berdasarkan **Teorema Lucas**, dapat disimpulkan bahwa 347adalah bilangan prima.

## C. Teori Grup

Selanjutnya, pada subbab ini dijelaskan beberapa konsep dasar struktur aljabar seperti grup, grup siklik, gelanggang, dan sebagainya. Konsep ini penting, karena pada pembahasan selanjutnya mengenai algoritma ElGamal,

serta tanda tangan digital dengan algoritma ElGamal, perhitungan-perhitungannya dilakukan di dalam suatu struktur aljabar.

## 1. Grup

Berikut, akan dijelaskan tentang struktur aljabar yang dinamakan dengan grup. Grup merupakan salah satu struktur aljabar yang berkenaan dengan suatu himpunan yang tidak kosong dan suatu operasi biner pada himpunan itu, serta memenuhi sifat-sifat, dimana akan dijelaskan sebagai berikut.

**Definisi 2.15 ( Sukirman, 2006 : 41)**

Misalkan $G$ adalah himpunan yang tak kosong dan operasi $*$ pada $G$ adalah suatu operasi biner. Himpunan $G$ bersama-sama dengan operasi biner $*$ atau ditulis $(G, *)$ adalah suatu grup, bila memenuhi aksioma-aksioma berikut, yaitu:

(i) Bersifat asosiatif

$\forall a, b, c \in G, \ (a * b) * c = a * (b * c)$

(ii) $G$ memuat elemen identitas, misal e

$\exists e \in G \ni \forall a \in G$ berlaku $a * e = e * a = a$

(iii)Setiap unsur $G$ mempunyai invers di dalam $G$ pula.

$\forall a \in G, \exists a^{-1} \in G$, $a^{-1}$ disebut invers dari $a$, sedemikian hingga $a * a^{-1} = a^{-1} * a = e$

Jika $(G, *)$ merupakan suatu grup yang memenuhi sifat komutatif, maka $(G, *)$ disebut dengan grup abelian atau grup komutatif. Banyaknya elemen grup $G$ disebut dengan order dari grup $G$. Sering ditulis dengan $\circ(G)$.

Suatu grup dengan operasi + disebut dengan grup aditif, dan grup dengan operasi × disebut dengan grup multiplikatif.

**Contoh 2.19:**

G={0, 1} dengan operasi biner penjumlahan dalam modulo 2 adalah sebuah grup. Operasi biner penjumlahan modulo 2 didefinisikan sebagai tabel *Cayley* berikut :

| + | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 0 |

*G* adalah suatu grup, karena memenuhi aksioma di atas.

(i) Tertutup, karena semua hasil operasi penjumlahan selalu menghasilkan nilai yang terdapat di dalam *G*.

(ii) Asosiatif, operasi penjumlahan modulo 2 bersifat asosiatif yang berarti bahwa $(a + b) + c = a + (b + c)$, untuk semua $a, b, c \in G$

(iii) Memiliki elemen identitas. Elemen 0 adalah elemen identitas dan memiliki sifat bahwa $a + 0 = 0 + a = a$.

(iv) Untuk semua elemen *a* di dalam G , elemen $0^{-1}$ adalah 0 dan elemen $1^{-1}$ adalah 1, sehingga 0 + 0 = 0, dan 1 + 1 = 0.

**Definisi 2.16 (Sukirman, 2006 : 49)**

Misalkan G suatu grup, $a \in G$ dan m suatu bilangan bulat positif, maka

$a^m = a\ a\ a\ \ldots\ a$ sebanyak *m* faktor

$a^{-m} = (a^{-1})^m$ dengan $a^{-1}$ adalah invers dari *a*

$a^0 = e$ (elemen identitas)

**Teorema 2.18 (Sukirman, 2006 :50)**

Misalkan G suatu grup, *m* dan *n* sembarang bilangan-bilangan bulat, maka $\forall a \in G$ berlaku

(i) $a^m a^n = a^{m+n}$

(ii) $(a^m)^n = a^{mn}$

**2. Grup Siklik**

**Definisi 2.17 (Sukirman, 2006 :79)**

Misalkan *G* suatu grup dan $a \in G$. Periode (order) *a* (disimbolkan $\circ(a)$ ) adalah suatu bilangan positif terkecil, misalnya *m*, sedemikian sehingga $a^m = e$. Apabila bilangan bulat positif *m* demikian itu tidak ada, maka dikatakan bahwa periode *a* adalah takhingga atau nol.

**Contoh 2.20** :

G ={1, 2, 4, 5, 7, 8} dengan $\times_9$ adalah suatu grup

$\circ(1) = 1$, $\circ(2) = 6$, sebab $2^6 = 1$, $\circ(4) = 3$, sebab $4^3 = 1$, $\circ(5) = 6$, $\circ(7) = 3$, dan $\circ(8) = 2$.

**Definisi 2.18 (Sukirman, 2006 :81)**

Grup *G* disebut grup siklik, apabila ada suatu elemen *G*, misalnya $a \in G$, sedemikian sehingga untuk setiap $x \in G$, $x = a^m$ untuk suatu bilangan bulat *m*. Selanjutnya, *a* disebut generator (elemen penghasil) dari *G* dan ditulis $G=(a)$.

**Contoh 2.21 :**

1. $G$ ={1, 2, 3, 4, 5, 6} dengan $\times_7$ adalah suatu grup siklik dengan generator 3 atau 5. Dikarenakan $1 = 3^6$, $2 = 3^2$, $3 = 3^1$, $4 = 3^4$, $5 = 3^5$, $6 = 3^3$ atau $1 = 5^6$, $2 = 5^4$, $3 = 5^5$, $4 = 5^2$, $5 = 5^1$, $6 = 5^3$.
2. Diberikan suatu grup $G$={1, $a$, $a^2$, $a^3$,..., $a^n$}. Generator dari grup tersebut adalah elemen yang memiliki pangkat yang saling prima dengan order G. Misalnya G={1, $a$, $a^2$, $a^3$,..., $a^{17}$}. Order G adalah 18, maka generator dari G adalah $a$, $a^5$, $a^7$, $a^{11}$, $a^{13}$, $a^{17}$.

**Teorema 2.19 (Sukirman, 2006 :82)**

Jika G merupakan grup berhingga memuat suatu elemen yang periodenya sama dengan order G, maka G adalah grup siklik dengan generator elemen tersebut.

**Teorema 2.20**

Jika G suatu grup berhingga, maka $\circ(a) \mid \circ(G)$ , $\forall\, a \in G$.

**Definisi 2.19(Menezes, Oorcshot, and Vanstone, 1996 : 69 )**

Grup multiplikatif dari $\mathbb{Z}_n$ adalah $\mathbb{Z}_n^* = \{\, a \in \mathbb{Z}_n \mid gcd(a, n) = 1\}$. Jika $p$ adalah bilangan prima, maka $\mathbb{Z}_p^* = \{\, a \mid 1 \leq a \leq p\text{-}1, a \in \mathbb{Z}\}$.

**Definisi 2.20 (Menezes, Oorcshot, and Vanstone, 1996 : 69 )**

Order dari $\mathbb{Z}_n^*$ didefinisikan sebagai banyaknya elemen pada $\mathbb{Z}_n^*$, dan dinotasikan dengan $|\mathbb{Z}_n^*|$

Berdasarkan Definisi 2.19 dan Definisi 2.20 di atas dapat disimpulkan bahwa order dari $\mathbb{Z}_p^*$ adalah $p\text{-}1$.

**Teorema 2.21**

Misalkan $p$ merupakan bilangan prima , maka himpunan $\mathbb{Z}_p^* = \{1, 2, 3, ..., p\text{-}1\}$ adalah suatu grup abelian berhingga dengan order $p$-1.

Dibawah ini akan dibuktikan bahwa $\mathbb{Z}_p^*$ merupakan grup abelian.

**Bukti :**

1. Akan dibuktikan $\mathbb{Z}_p^*$ bersifat tertutup terhadap operasi perkalian modulo $p$.

   Bukti :

   Ambil dua sebarang $a$, $b \in \mathbb{Z}_p^*$, maka $1 \leq a \leq p$ -1, dan $1 \leq b \leq p$ -1. Misalkan $r \equiv a.b \pmod{p}$ dengan $r$ adalah residu terkecil tak negatif modulo $p$, maka $r \leq p$ -1. Diketahui $p$ bilangan prima maka $p \nmid ab$. Jadi $r \neq 0$, sehingga $1 \leq r \leq p$ -1, dengan demikian $r \equiv a.b \pmod{p}$, $r \in \mathbb{Z}_p^*$. Jadi terbukti bahwa $\mathbb{Z}_p^*$ tertutup terhadap operasi perkalian modulo $p$.

2. Akan dibuktikan $\mathbb{Z}_p^*$ bersifat asosiatif terhadap operasi perkalian modulo $p$.

   Bukti :

   Ambil tiga sebarang elemen $\mathbb{Z}_p^*$, misal $a$, $b$, $c \in \mathbb{Z}_p^*$, sedemikian hingga $r \equiv (a.b).c \pmod{p}$ atau $kp + r = (a.b).c$ untuk $k$ anggota bilangan bulat. Karena $a$, $b$, c bilangan bulat maka berlaku sifat asosiatif, yaitu $(a.b).c = a.(b.c)$, akibatnya $kp + r = a.(b.c)$ atau $r \equiv a.(b.c) \pmod{p}$. Sehingga dapat disimpulkan bahwa $\mathbb{Z}_p^*$ bersifat asosiatif terhadap operasi perkalian modulo $p$.

3. Dibuktikan $\mathbb{Z}_p^*$ memiliki elemen identitas terhadap operasi perkalian modulo $p$.

   Bukti :

Misalkan $b$ adalah elemen identitas di $\mathbb{Z}_p^*$ sedemikian hingga $ab \equiv a \pmod{p}$, karena $gcd(a, p) = 1$ maka $b \equiv 1 \pmod{p}$. Dengan demikian bilangan bulat $1 \in \mathbb{Z}_p^*$ adalah elemen identitas $\mathbb{Z}_p^*$, jadi $\mathbb{Z}_p^*$ memiliki elemen identitas 1 terhadap operasi perkalian modulo $p$.

4. Dibuktikan setiap elemen $\mathbb{Z}_p^*$ memiliki elemen invers terhadap operasi perkalian modulo $p$.

   Bukti :

   Ambil sebarang elemen $a \in \mathbb{Z}_p^*$. Diketahui $gcd(a, p) = 1$, maka ada bilangan bulat $x \in \mathbb{Z}_p^*$ sedemikian hingga $ax = 1 \pmod{p}$, perkongruenan tersebut mempunyai satu solusi, yaitu bilangan bulat $x$ atau $a^{-1} = x \pmod{p}$. Jadi untuk setiap $a \in \mathbb{Z}_p^*$ memiliki elemen invers $a^{-1} \in \mathbb{Z}_p^*$ terhadap operasi perkalian modulo $p$.

5. Dibuktikan $\mathbb{Z}_p^*$ bersifat komutatif terhadap perkalian modulo $p$.

   Bukti :

   Ambil sebarang $a, b \in \mathbb{Z}_p^*$. Misalkan $r \equiv a.b \pmod{p}$ dengan $r$ adalah residu terkecil tak negatif modulo $p$, sedemikian hingga $kp + r = ab$, untuk $k$ anggota bilangan bulat. Karena $a$, $b$ merupakan anggota bilangan bulat maka berlaku sifat komutatif, sehingga $kp + r = ba \Rightarrow r \equiv b.a \pmod{p}$. Jadi terbukti bahwa $\mathbb{Z}_p^*$ bersifat komutatif terhadap operasi perkalian modulo $p$.

   Dari pembuktian diatas dapat disimpulkan bahwa $\mathbb{Z}_p^*$ adalah grup abelian. ■

Selanjutnya, akan dibuktikan bahwa $\mathbb{Z}_p^*$ merupakan grup siklik.

Ambil $a \in \mathbb{Z}_p^*$, dan $m$ merupakan periode maksimum dari $a$. Berdasarkan Definisi 2.17 diperoleh $a^m = e = 1$ (elemen identitas $\mathbb{Z}_p^* = 1$ ), $a^m - 1 = 0$ . Setiap elemen tak nol dari $\mathbb{Z}_p^*$ merupakan akar persamaan dari $a^m - 1 = 0$, dan persamaan $a^m - 1 = 0$ memiliki paling banyak $m$ persamaan, sehingga $m \geq p$-1. Menurut Teorema 2.20 , $m \mid p$-1. Jadi, dapat disimpulkan bahwa $m = p$- 1 . Dengan demikian periode dari $a = p - 1$. Berdasarkan Definisi 2.19, terbukti bahwa $\mathbb{Z}_p^*$ merupakan grup siklik. ■

**Definisi 2.21 (Buchmann, 2000 : 63)**

Suatu elemen yang membangun $\mathbb{Z}_p^*$ disebut *elemen primitif* (*primitive root*) *mod p*.

**Akibat 2.21 (Sukirman, 2006 :220)**

Setiap bilangan prima *p* mempunyai $\phi(p\text{-}1)$ akar primitif.

**Contoh 2.22 (Buchmann, 2000 : 63)**

Diberikan $p = 13$. Karena 13 adalah bilangan prima, maka menggunakan Akibat 2.21 diperoleh $\phi(13) = 12 - 1 = 12$ . Kemudian dihitung order dari masing-masing elemen $\mathbb{Z}_{13}^*$. Diperoleh empat elemen yang mempunyai order 12 dan sama dengan order dari $\mathbb{Z}_{13}^*$ yaitu 12. Oleh karena elemen yang mempunyai order 12 adalah pembangun, maka elemen tersebut adalah elemen primitif. Empat elemen tersebut adalah 2, 6, 7 dan 11. Sehingga, elemen primitif $\mathbb{Z}_{13}^*$ adalah 2, 6, 7dan 11.

**Teorema 2.22 (Buchmann, 2000 : 41)**

Diberikan $G$ adalah suatu grup, dan $g \in G$, $l$ dan $h$ adalah bilangan bulat. Maka $g^l = g^k$ jika dan hanya jika $l \equiv k \mod (\text{order } g)$.

**3. Gelanggang**

**Definisi 2.22 (Fraleigh, 2000 : 167 )**

Suatu gelanggang $(R, +, .)$ adalah himpunan tak kosong yang dilengkapi dengan dua operasi biner, yaitu " +" (penjumlahan) dan "." (perkalian) , yang memenuhi

1. $(R, +)$ suatu grup abelian yaitu,
   (i) Sifat tertutup terhadap penjumlahan
   $\forall\, a, b \in R, a + b \in R$
   (ii) Sifat asosiatif terhadap penjumlahan
   $\forall\, a, b, c \in R, (a+b) + c = a+(b+c)$
   (iii) R memuat elemen identitas terhadap penjumlahan
   $\exists\, z \in R, \forall\, a \in R, a + z = z+a = a$
   (iv) Setiap elemen R memiliki invers terhadap penjumlahan
   $\forall\, a \in R, \exists\, (-a) \in R, a + (-a) = (-a)+ a = z$
   (v) Sifat komutatif penjumlahan
   $\forall\, a, b \in R, a + b = b + a$
2. $(R, .)$ bersifat semigrup, yaitu
   (i) Sifat tertutup terhadap perkalian
   $\forall\, a, b \in R, (a.b) \in R$

(ii) Sifat asosiatif terhadap perkalian

$\forall\, a\,,\ b, c\ \in R.\ (a.b).c = a.(b.c)$

3. Untuk setiap $a, b, c \in R$ berlaku sifat distributif kiri, yaitu $a.(b+c) = a.b + a.c$ dan sifat distributif kanan yaitu $(a+b).c = a.c + b.c$

**Contoh 2.23**

Diberikan suatu himpunan seluruh bilangan bulat dengan dilengkapi dua buah operasi biner, yaitu "+" dan ".", yang dituliskan $(\mathbb{Z}, +, .)$ adalah suatu gelanggang.

**Definisi 2.23 (Sukirman, 2006 :2 )**

Jika $(R, +, .)$ adalah suatu gelanggang dan mempunyai sifat komutatif terhadap perkalian, yaitu

$$\forall\, a, b\ \in \mathbb{Z}\,, \qquad a.b = b.a$$

maka $(R, +, .)$ dinamakan gelanggang komutatif (gelanggang abelian).

**Definisi 2.24 (Sukirman, 2006 :2 )**

Jika $(R, +, .)$ adalah suatu gelanggang dan jika ada $u \in \mathbb{Z}$ sedemikian sehingga

$$\forall\, a \in \mathbb{Z}\,, \qquad a.u = u.a = a$$

Maka $u$ disebut dengan elemen kesatuan dan $(R, +, .)$ disebut gelanggang dengan elemen kesatuan.

**Contoh 2.24**

$\mathbb{Q}$ adalah himpunan semua bilangan rasional dan dilengkapi dengan operasi $\oplus$ dan $\odot$ pada $\mathbb{Q}$ didefinisikan sebagai berikut.

$\forall\, a\,,\ b \in \mathbb{Q}\,,\ a \oplus b = a + b + 1\,,\ a \odot b = a + a.b + b$

Selanjutnya akan dibuktikan $(\mathbb{Q}, \oplus, \odot)$ adalah sebuah **gelanggang komutatif dengan elemen kesatuan**.

1) Ditunjukkan bahwa $(\mathbb{Q}, \oplus)$ adalah grup abelian.

(i) Ambil $a$ , $b \in \mathbb{Q}$ , $a \oplus b = a + b + 1$, dan $(a+b+1) \in \mathbb{Q}$ (memenuhi sifat tertutup terhadap penjumlahan)

(ii) Ambil $a$ , $b$ , $c \in \mathbb{Q}$ , maka

$$(a \oplus b) \oplus c = (a + b + 1) \oplus c$$
$$= (a + b + 1) + c + 1$$
$$= a + (b + c + 1) + 1$$
$$= a \oplus (b + c + 1)$$
$$= a \oplus (b \oplus c)$$

Jadi memenuhi sifat asosiatif terhadap penjumlahan.

(iii) Misalkan z adalah elemen identitas dari $\mathbb{Q}$. Maka $\exists\, a \in \mathbb{Q}$, berlaku

$a \oplus z = z \oplus a = a$, sehingga

$a \oplus z = a + z + 1 = a$, diperoleh $z = -1$

Jadi elemen identitas dari $\mathbb{Q}$ adalah -1.

(iv) Ambil $a, t \in \mathbb{Q}$, dimana $t = -a$ , berlaku

$a \oplus t = -1$, sehingga

$$a + t + 1 = -1$$
$$t = -(a + 2)$$

Jadi setiap elemen dari $\mathbb{Q}$ memiliki invers, yaitu t = -($a$+2)

(v) Ambil $a$ , $b \in \mathbb{Q}$ ,

$a \oplus b = a + b + 1 = b + a + 1 = b \oplus a$

Terbukti bahwa memenuhi sifat komutatif terhadap penjumlahan.

2) Ditunjukkan bahwa $(\mathbb{Q}, \odot)$ merupakan semigrup

(i) Ambil $a$ , $b \in \mathbb{Q}$ , $a \odot b = a + ab + b$, dan $(a+ab+b) \in \mathbb{Q}$

(memenuhi sifat tertutup terhadap perkalian)

(ii) Ambil $a$ , $b$ , $c \in \mathbb{Q}$ , maka

$(a \odot b) \odot c = (a + ab + b) \odot c$

$= (a + ab + b) + (a + ab + b)c + c$

$= a + ab + b + ac + abc + bc + c$..... ( I )

dan

$a \odot (b \odot c) = a \odot (b + bc + c)$

$= a + a(b + bc + c) + (b + bc + c)$

$= a + ab + abc + ac + b + bc + c$.....( II)

Berdasarkan (I) dan (II) disimpulkan bahwa $(a \odot b) \odot c = a \odot (b \odot c)$

(iii) Memenuhi sifat komutatif terhadap perkalian.

Ambil $a$ , $b \in \mathbb{Q}$, maka

$a \odot b = a + ab + b = b + ba + a = b \odot a$

3) Memenuhi sifat distributif kiri dan sifat distributif kanan $\odot$ terhadap $\oplus$.

Ambil $a$ , $b$ , $c \in \mathbb{Q}$ ,

(i) Sifat distributif kiri

$a \odot (b \oplus c) = a \odot (b + c + 1)$

$$= a + a(b + c + 1) + (b + c + 1)$$

$$= a + ab + ac + a + b + c + 1$$

$$= (a + ab + b) + (a + ac + c) + 1$$

$$= (a \odot b) \oplus (a \odot c)$$

(ii) Sifat distributif kanan

$$(a \oplus b) \odot c = (a + b + 1) \odot c$$

$$= (a + b + 1) + (a + b + 1)c + c$$

$$= a + b + 1 + ac + bc + c + c$$

$$= (a + ac + c) + (b + bc + c) + 1$$

$$= (a \odot c) \oplus (b \odot c)$$

4) Memiliki elemen kesatuan ($u$)

Misalkan elemen kesatuan dari $\mathbb{Q}$ adalah $u$ , maka

$a \odot u = u \odot a = a$, sehingga

$$a + au + u = a$$

$$(a + 1)u = 0$$

$$u = 0$$

Jadi elemen kesatuan dari $\mathbb{Q}$ adalah $u = 0$ .

Berdasarkan 1) , 2), 3), dan 4) terbukti bahwa $(\mathbb{Q}, \oplus, \odot)$ adalah sebuah gelanggang komutatif dengan elemen kesatuan. ■

**Definisi 2.25 (Sukirman, 2006 :16 )**

Jika ($R$, + , . ) adalah suatu gelanggang komutatif dengan elemen kesatuan, dan setiap elemen tak nolnya memiliki invers terhadap perkalian disebut dengan medan/*field* ($F$).

Suatu medan yang memuat elemen sebanyak berhingga, disebut dengan medan berhingga (*finite field*).

**Contoh 2.25**

Berdasarkan Contoh 2.24 , maka akan ditunjukkan bahwa $(\mathbb{Q}, \oplus, \odot)$ merupakan suatu medan.

Contoh 2.24 sudah terbukti bahwa $(\mathbb{Q}, \oplus, \odot)$ suatu gelanggang komutatif dengan elemen kesatuan, tinggal ditunjukkan bahwa setiap elemen tak nol dari $(\mathbb{Q}, \oplus, \odot)$ memiliki invers terhadap perkalian.

Ambil $a \in \mathbb{Q}$ , dengan $a \neq$z = -1. Misalkan $b \in \mathbb{Q}$ , dan b merupakan invers dari a terhadap $\odot$, maka

$$a \odot b = b \odot a = u$$

$$a \odot b = u = 0$$

$$a + ab + b = 0$$

$$b(a + 1) = -a$$

$b = \frac{-a}{(a+1)} \in \mathbb{Q}$ , $a + 1 \neq 0$, karena $a \neq -1$.

Jadi , $(\mathbb{Q}, \oplus, \odot)$ merupakan suatu medan.

## D. Kriptografi

Pada sub bahasan ini, akan dibahas tentang definisi kriptografi, terminologi kriptografi, tujuan dari kriptografi, dan jenis kriptografi. Sehingga akan memberikan penjelasan-penjelasan yang nantinya akan banyak dibahas pada Bab III.

## 1. Definisi Kriptografi

**Definisi 2.26 ( Menezes, 1996 :4)**

Kriptografi (*cryptography*) berasal dari Bahasa Yunani, yaitu *cryptos* yang berarti *secret* (rahasia), sedangkan *graphien* artinya *writing* (tulisan). Jadi secara asal bahasa kriptografi berarti *secret writing* (tulisan rahasia). Kriptografi memiliki beberapa definisi. Salah satu definisi kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, serta otentifikasi .

## 2. Terminologi Kriptografi

Di dalam kriptografi, akan sering ditemukan berbagai istilah (terminologi). Adapun istilah-istilah yang kerap kali digunakan adalah sebagai berikut.

*a.* Pesan, *Plaintext*, dan *Ciphertext*

Pesan adalah data ataupun suatu informasi yang dapat dibaca dan dimengerti maknanya. Dan nama lain untuk pesan ialah *plaintext*, atau teks jelas. *Ciphertext* adalah suatu bentuk pesan yang bersandi. Disandikannya suatu pesan adalah agar pesan tersebut tidak dapat dimengerti oleh pihak lainnya.

**Contoh 2.26.a (contoh *plaintext*)**

| |
|---|
| Ketika saya berjalan – jalan di pantai, saya menemukan banyak sekali kepiting yang merangkak menuju ke laut. Mereka adalahn anak kepiting yang baru saja menetas dari dalam pasir. |

**Contoh 2.26.b (contoh *ciphertext*)**

| |
|---|
| Κετικα σαψα βερφαλαν  φαλαν δι πανταi, σα ψα μενεμυκαν βανψακ σεκαλι κεπιτινγ ψανγ με ρανγκακ μενυφυ κε λαυτ. Μερεκα αδαλαην ανα κ κεπιτινγ ψανγ βαρυ σαφα μενετασ δαρι δαλαμ πασιρ. |

b. Pengirim dan Penerima

Suatu aktivitas komunikasi data, akan melibatkan pertukaran antara dua entitas, yakni pengirim dan penerima. Pengirim adalah entitas yang mengirim pesan kepada entitas lainnya. Sedangkan penerima adalah entitas yang menerima pesan. (Rinaldi, 2006 : 4). Suatu pengiriman pesan, pengirim tentu menginginkan pesan dapat dikirim secara aman. Untuk mengamankannya, pengirim biasanya akan menyandikan pesan yang dikirimkan tersebut.

c. Enkripsi dan Dekripsi (Rinaldi, 2006 : 4)

Suatu proses untuk menyandikan *plaintext* menjadi *ciphertext* disebut enkripsi (*encryption*). Sedangkan proses pengembalian dari *ciphertext* manjadi *plaintext* dinamakan dekripsi (*decription*). Enkripsi dan dekripsi merupakan suatu pesan yang memetakan elemen-elemen antara kedua himpunan tersebut. Misalkan $P$ adalah himpunan *plaintext*, dan $C$ adalah himpunan *ciphertext*, maka fungsi enkripsi $E$ memetakan $P$ ke $C$, ditulis $E(P) = C$. Dan fungsi dekripsi $D$ memetakan $C$ ke $P$, ditulis $D(C) = P$.

d. *Cipher* dan Kunci

Algoritma kriptografi disebut juga *cipher* yaitu aturan atau fungsi matematika yang digunakan untuk enkripsi dan dekripsi. Beberapa *cipher* memerlukan algoritma yang berbeda untuk enkripsi dan dekripsi (Stalling, 2005: 30).

Untuk menjaga kerahasiaan pengiriman pesan dalam kriptografi modern dibutuhkan kunci. Kunci (*key*) adalah parameter yang digunakan untuk mentransformasi proses pengenkripsian dan pendekripsian pesan. Biasanya, kunci berupa deretan bilangan maupun string. Dengan menggunakan kunci *K* maka proses enkripsi dan dekripsi dapat ditulis sebagai $E_K(P) = C$ dan $D_K(C) = P$, dan kedua fungsi tersebut memenuhi $D_K(E_K(P)) = P$

e. Sistem Kriptografi Kunci Simetri dan Tak Simetri

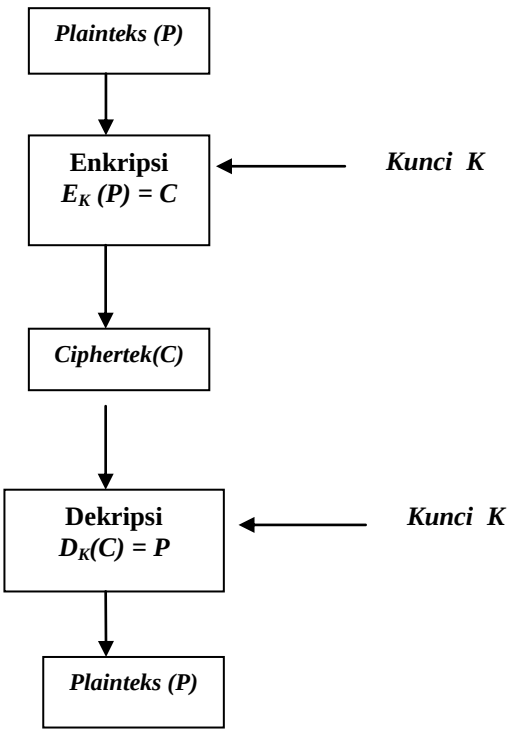
Sistem kriptografi merupakan kumpulan yang terdiri dari *plaintext*, *ciphertext*, kunci, enkripsi serta dekripsi.(Stinson, 2006 :1)

Berdasarkan kunci yang digunakan dalam proses enkripsi dan dekripsi, kriptografi dapat dibedakan menjadi kriptografi kunci simetri dan kriptografi kunci tak simetri. Kriptografi kunci tak simetri ini sering disebut dengan kriptografi kunci publik.

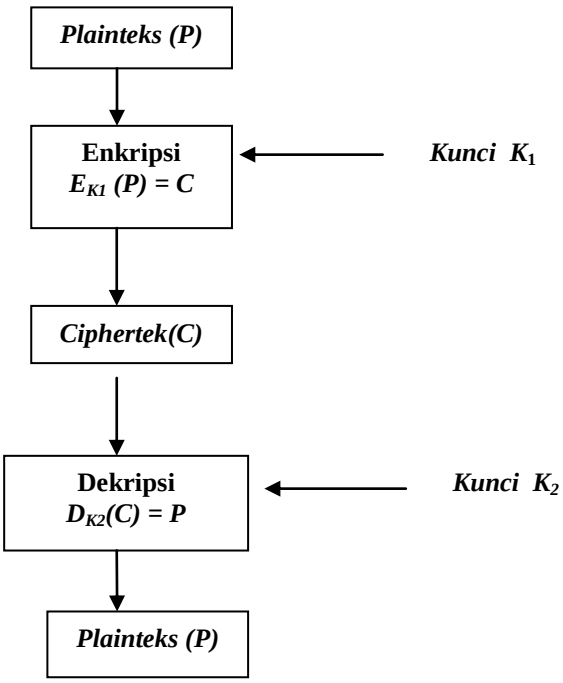
Kriptografi kunci simetri, sering disingkat menjadi kriptografi simetri, kunci yang digunakan pada proses enkripsi dan dekripsi adalah sama. Oleh karena itu, sebelum saling berkomunikasi kedua belah pihak harus melakukan kesepakatan dalam menentukan kunci yang akan

digunakan. Keamanan menggunakan sistem ini terletak pada kerahasiaan kunci yang akan digunakan. Sedangkan dalam sistem kriptografi kunci publik, kunci yang digunakan dalam proses enkripsi dan dekripsi berbeda. Sistem ini terdapat dua buah kunci, yaitu kunci publik dan kunci privat. Kunci publik digunakan untuk proses enkripsi, dan kunci privat digunakan untuk mendekripsikan pesan. Kunci publik bersifat tak rahasia, sedangkan kunci privat hanya boleh diketahui oleh penerima pesan. Salah satu contoh algoritma kriptografi kunci publik ini adalah ElGamal, yang nantinya akan dibahas pada Bab III.

Dibawah ini akan diberikan gambar tentang skema kriptografi simetri dan kriptografi kunci publik




**Gambar 2.1 . Skema Kriptografi Simetri (Rinaldi, 2006 : 14)**




**Gambar 2.2 . Skema Kriptografi Kunci Publik (Rinaldi, 2006 : 14)**

### 3. Tujuan Kriptografi (Menezes, 1996 :4)

Tujuan dari kriptografi adalah sebagai berikut.

a. Kerahasiaan (*confidentiality*), merupakan suatu layanan yang digunakan untuk menjaga isi dari informasi dari pihak-pihak yang tak berhak untuk mendapatkannya.

b. Integritas Data (*data integrity*), merupakan suatu layanan dimana menjamin bahwa pesan masih asli, dan belum dimanipulasi oleh pihak - pihak yang tidak berhak. Realisasi layanan ini di dalam kriptografi, adalah dengan menggunakan tanda tangan digital.

c. Otentifikasi (*authentication*), merupakan suatu layanan yang berhubungan dengan identifikasi. Misalnya, mengidentifikasi suatu kebenaran pihak-pihak yang berkomunikasi (entitas) maupun

mengidentifikasi kebenaran sumber pesan. Sama seperti poin (b), di dalam kriptografi, layanan ini diwujudkan dengan menggunakan tanda tangan.

d. Nirpenyangkalan (*non-repudiation*), merupakan suatu layanan untuk mencegah entitas yang saling berkomunikasi melakukan penyangkalan. Misalkan salah satu dari entitas menyangkal telah mengirim maupun menerima pesan.

# BAB III

# PEMBAHASAN

Sistem kriptografi ElGamal merupakan suatu sistem kriptografi yang menggunakan algoritma ElGamal. Algoritma ElGamal merupakan algoritma kriptografi asimetris. Yang pertama kali ditemukan oleh Taher Gamal pada tahun 1984. Algoritma ini didasarkan pada masalah logaritma diskret pada grup $\mathbb{Z}_p^*$.

## A. Masalah Logaritma Diskret

Sebelum membahas tentang sistem kriptografi ElGamal, akan dijelaskan tentang masalah logaritma diskret. Misalkan G adalah suatu grup siklik dengan order *n*, $\alpha$ adalah pembangun G dan elemen identitas dari G adalah 1. Diberikan $\gamma \in$ G. Masalah yang dimunculkan ialah bagaimana menentukan suatu bilangan bulat nonnegatif terkecil *b* sedemikian sehingga memenuhi

$$\gamma \equiv \alpha^b .$$

Bilangan bulat *b* seperti ini disebut dengan logaritma diskret dari $\gamma$ dengan basis $\alpha$. Masalah bagaimana untuk menentukan bilangan bulat *b* seperti ini disebut dengan masalah logaritma diskret.

Masalah komputasi logaritma diskret sangat penting dalam kriptografi. Banyak kegiatan kriptografi yang tumpuan keamanannya menggunakan masalah logaritma diskret. Misalnya digunakan sebagai dasar pembangkitan kunci pada sistem kriptografi ElGamal.

## B. Sistem Kriptografi ElGamal

Pada algoritma ElGamal ini terdiri dari tiga proses, yaitu proses pembangkitan pasangan kunci , proses enkripsi, dan proses dekripsi. Algoritma ini melakukan proses enkripsi pada blok-blok *plaintext* dan kemudian menghasilkan blok-blok *ciphertext* yang kemudian dilanjutkan dengan proses dekripsi, dimana hasilnya digabungkan kembali,sehingga menjadi pesan yang utuh dan mudah dipahami. Untuk pembentukan sistem kriptografi ElGamal, dibutuhkan bilangan prima *p* dan elemen primitif yang membentuk grup $\mathbb{Z}_p^*$.

### 1. Proses Pembentukan Kunci

Proses pertama adalah pembentukan kunci yang terdiri dari kunci rahasia dan kunci publik. Pada proses ini dibutuhkan sebuah bilangan prima $p$ yang digunakan untuk membentuk grup $\mathbb{Z}_p^*$, elemen primitif $g$ dan sebarang $a \in \{1,2,3,\dots,p-2\}$. Dipilih $a$ pada rentang tersebut karena dalam ElGamal merupakan bilangan yg digunakan untuk operasi pangkat, padahal diketahui grup $\mathbb{Z}_p^*$ berorder p-1, oleh karena itu pangkatnya dari {1,2,...,p-2}. Jika dipangkarkan dengan *p*-1 maka akan menghasilkan elemen identitas, dan pada grup $\mathbb{Z}_p^*$ elemen identitasnya adalah 1. Untuk kunci publik pada algoritma ElGamal ini adalah tiga pasangan bilangan yaitu *( p, g,y)* , dengan

$$y = g^a mod\ p \quad \text{.....(3.1)}$$

Sedangkan untuk kunci privatnya adalah *a* .

Karena algoritma ElGamal menggunakan bilangan bulat dalam proses perhitungannya, maka pesan yang akan dikirimkan harus dikonversikan ke dalam bilangan bulat. Sebagai pengkonversiannya,

digunakan kode ASCII (*American Standard for Information Interchange*). Kode ini adalah representasi numerik dari karakter yang digunakan pada komputer, yang nilai minimalnya 0 dengan nilai maksimalnya 255. Berdasarkan sistem kriptografi ElGamal, maka bilangan prima yang digunakan adalah lebih besar dari 255. Kode ASCII berkorespondensi satusatu dengan karakter pesan yang akan dikirimkan. Di bawah ini akan dituliskan algoritma pembentukan kunci .

**Algoritma 3.1. Pembentukan Kunci**

Input : bilangan prima $p > 255$ , dan elemen primitif $g$, dimana $g < p$ .

Output : kunci publik ( $p, g, y$), dan kunci rahasia $a$.

Langkah :

1. Pilih bilangan prima $p$.
2. Pilih dua buah bilangan acak $g$ dan $a$, dengan $a \in \{1,2,3, \dots, p-2\}$
3. Hitung $y = g^a mod\ p$
4. Publikasikan nilai $p$, $g$ ,dan $y$, tetapi nilai $a$ dirahasiakan.

Kunci publik yang dihasilkan pada pembentukan kunci ini bersifat tunggal, karena ketiga nilai yang akan digunakan pada pembentukan kunci sudah ditetapkan, sehingga nilai $y$ adalah tunggal.

Terdapat $y_1$ dan $y_2$. Diambil bilangan prima $p$, bilangan acak $g \in \mathbb{Z}_p^*$, dan $a \in \{1,2,3, \dots, p-2\}$. Dengan $y = g^a \bmod p$, maka

$y_1 = g^a \bmod p$ dan $y_2 = g^a \bmod p$

$y_1 = g^a \bmod p$, maka

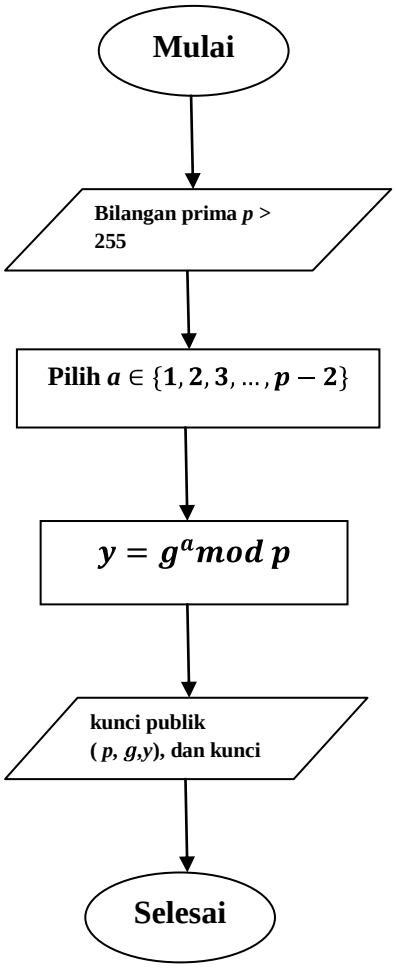
$g^a = y_1 \mod p$ (***)

(***) disubstitusikan ke dalam persamaan (**), sehingga

$y_2 = y_1 \mod p$

dengan demikian $y_2 = y_1$ , terbukti bahwa kunci $y$ bersifat tunggal.

Berdasarkan Algoritma 3.1 , dapat dibuat suatu diagram alir tentang proses pembentukan kunci, yaitu seperti di bawah ini.




**Gambar 3.1. Diagram Alir Pembentukan Kunci**

Pihak yang melakukan proses pembuatan kunci adalah pihak penerima. Jadi pihak penerima mengetahui kunci publik dan kunci privat. Kunci publik yang dia hasilkan, diberitahukan kepada pengirim pesan. Namun, pengirim pesan tidak mengetahui kunci privatnya. Berikut ini akan diberikan contoh proses pembentukan kunci dengan menggunakan algoritma ElGamal.

Contoh 3.1 :

Misalkan Ilham dan Rizal adalah rekan kerja. Suatu saat, Ilham akan mengirimkan pesan rahasia kepada Rizal. Oleh karena itu, Rizal harus membuat kunci publik dan kunci privatnya. Rizal memilih bilangan prima $p$ = 2357, dan elemen primitif $g$ =2. Selanjutnya dipilih $a$ = 1751. Lalu dihitung

$$y = 2^{1751} mod\ 2357 = 1185$$

Jadi diperoleh kunci publik ( $p, g, y$) = ( 2357, 2, 1185), dan kunci privat $a$ = 1751. Rizal memberikan kunci publik ( 2357, 2, 1185) kepada Ilham, dan untuk kunci privatnya tetap ia simpan sendiri.

**2. Proses Enkripsi**

*Plaintext* adalah himpunan dari {0, 1, 2 , ..., $p$-1}.Untuk mengenkripsi sebuah *plaintext* $m$ , dibutuhkan kunci publik ( $p, g, y$) yang sebelumnya telah dibuat oleh penerima pesan. Lalu dipilih sebarang bilangan acak rahasia $k$ dimana $k \in \{1,2,3, \dots, p-2\}$. Pesan yang akan disampaikan adalah $m$ , lalu m akan dipecah tiap-tiap karakternya, yang dikonversikan ke dalam kode ASCII, sehingga pesan menjadi *plaintext* $m_1, m_2, m_{2,\dots,} m_n$ dengan

$m_1 \in \{1,2,3,\dots,p-1\}$, $i$ = 1,2,...,*n*. Lalu proses pengenkripsian dilakukan pada tiap blok-blok *m* dengan menghitung

$$B = g^k \bmod p \quad \text{.....} \quad (3.2)$$

dan

$$\beta = y^k m \bmod p \text{.....} \quad (3.3)$$

dengan $k \in \{1,2,3,\dots,p-2\}$ acak., sehingga nanti akan diperoleh *ciphertext* $(B,\beta)$ untuk blok pesan *m*. Jadi ukuran *ciphertext* dua kali ukuran *plaintext*.

Proses penentuan bilangan acak *k*, pengirim pesan yang berperan, dan sifat bilangan acak *k* tadi adalah rahasia, jadi hanya pengirim pesan saja yang mengetahuinya.

.

**Algoritma 3.2. Proses Enkripsi**

Input : Pesan yang akan dikirimkan, kunci publik ( $p$, $g$, $y$) .

Output : *Ciphertext* $(B_i, \beta_i)$, $i$ = 1,2,...,*n*.

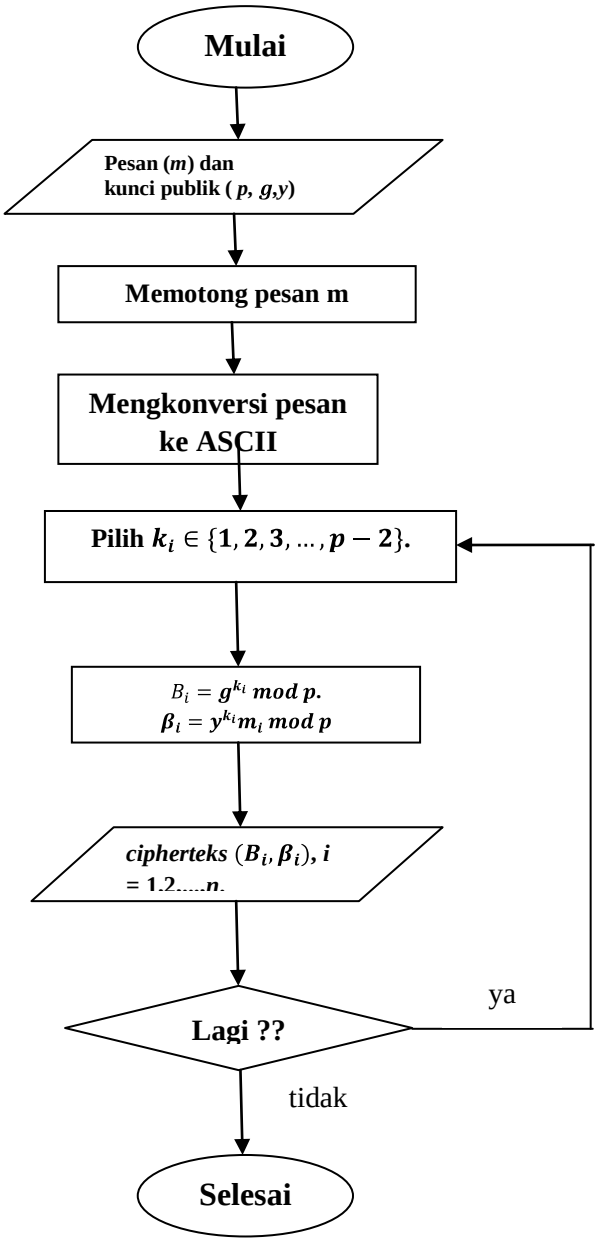
Langkah :

1. Memotong pesan *m* menjadi blok-blok pesan, sehingga satu blok adalah satu karakter pesan.
2. Mengkonversikan masing-masing karakter yang telah diperoleh ke dalam kode ASCII, sehingga diperoleh *plaintext* sebanyak *n* bilangan, yaitu $m_1$, $m_2$ ,..., $m_n$ .
3. Untuk *i* dari 1 sampai *n* :
   a. Memilih sebarang bilangan acak $k_i \in \{1,2,3,\dots,p-2\}$.
   b. Menghitung $B_i = g^{k_i} \bmod p$.

c. Menghitung $\beta_i = y^{k_i} m_i \ mod \ p$ .

4. Diperoleh *ciphertext* $(B_i, \beta_i)$, $i$ = 1,2,...,$n$.

Berdasarkan Algoritma 3.2, di bawah ini akan diberikan suatu diagram alir proses enkripsi suatu pesan dengan menggunakan sistem kriptografi ElGamal.




**Gambar 3.2. Diagram Alir Proses Enkripsi Pesan**

Contoh 3.2:

Dari contoh sebelumnya, Ilham memperoleh kunci publik ($p$, $g$,$y$) = (2357, 2, 1185). Ilham dan Rizal adalah teman kantor, Ilham ingin menyampaikan laporan kerja perusahaan kepada Rizal. Karena Ilham sedang berada di luar, Rizal harus membukanya langsung pada komputer milik Ilham . Oleh karena itu Ilham mengirimkan pesan rahasia kepada Rizal, yang isinya adalah "Password file=220409 ". Karena pesan bersifat rahasia maka pesan tersebut dienkripsi terlebih dahulu oleh Ilham dengan menggunakan kunci publik (2357,2,1185) yang sebelumnya diberikan Rizal. Lalu Ilham akan melakukan proses enkripsi. Langkah pertama ia harus memotong pesan menjadi blok-blok karakter yang kemudian ia konversikan ke dalam kode ASCII. Untuk hasil pengkonversiannya dapat dilihat pada tabel berikut ini.

**Tabel 3.1. Konversi Karakter Pesan Ke Kode ASCII**

| ***i*** | $m_i$ | **Karakter** | **ASCII** |
|---|---|---|---|
| 1 | $m_1$ | P | 80 |
| 2 | $m_2$ | a | 97 |
| 3 | $m_3$ | s | 115 |
| 4 | $m_4$ | s | 115 |
| 5 | $m_5$ | w | 119 |
| 6 | $m_6$ | o | 111 |
| 7 | $m_7$ | r | 114 |
| 8 | $m_8$ | d | 100 |
| 9 | $m_9$ | <spasi> | 32 |
| 10 | $m_{10}$ | f | 102 |
| 11 | $m_{11}$ | i | 105 |
| 12 | $m_{12}$ | l | 108 |

| | | | |
|---|---|---|---|
| 13 | $m_{13}$ | e | 101 |
| 14 | $m_{14}$ | = | 61 |
| 15 | $m_{15}$ | 2 | 50 |
| 16 | $m_{16}$ | 2 | 50 |
| 17 | $m_{17}$ | 0 | 48 |
| 18 | $m_{18}$ | 4 | 52 |
| 19 | $m_{19}$ | 0 | 48 |
| 20 | $m_{20}$ | 9 | 57 |

Dapat dilihat pada tabel di atas, bahwa banyaknya blok (karakter) dari pesan tersebut adalah *n* = 20. Lalu selanjutnya, menentukan bilangan acak *k* dimana $k_i \in \{1,2,3,\dots,2355\}$,$i$ =1,2,...,20. Setelah itu mencari nilai *B* dengan menggunakan rumus (3.2) , yaitu $B_i = 2^{k_i} \, mod \, 2357$. Selanjutnya, dengan menggunakan rumus (3.3), menhitung $\beta_i = 1185^{k_i} \, m_i \, mod \, 2357$, dengan *i*=1,2,...,20. Lebih jelasnya hasil enkripsi disajikan dalam tabel di bawah ini.

**Tabel 3.2 . Proses Enkripsi**

| $i$ | $m_i$ | $k_i$ | $B_i = 2^{k_i} \, mod \, 2357$ | $\beta_i = 1185^{k_i} \, m_i \, mod \, 2357$ |
|---|---|---|---|---|
| 1 | 80 | 141 | 955 | 140 |
| 2 | 97 | 1527 | 1551 | 2084 |
| 3 | 115 | 1823 | 930 | 1264 |
| 4 | 115 | 2175 | 432 | 363 |
| 5 | 119 | 1208 | 1311 | 559 |
| 6 | 111 | 978 | 1732 | 290 |
| 7 | 114 | 189 | 2183 | 243 |
| 8 | 100 | 1061 | 1345 | 128 |
| 9 | 32 | 2204 | 2021 | 551 |

| | | | | |
|---|---|---|---|---|
| 10 | 102 | 879 | 609 | 459 |
| 11 | 105 | 2108 | 51 | 842 |
| 12 | 108 | 1610 | 754 | 1352 |
| 13 | 101 | 1902 | 2008 | 894 |
| 14 | 61 | 245 | 803 | 2041 |
| 15 | 50 | 2307 | 1616 | 548 |
| 16 | 50 | 1404 | 2281 | 432 |
| 17 | 48 | 1406 | 2053 | 2060 |
| 18 | 52 | 506 | 2168 | 961 |
| 19 | 48 | 2010 | 1971 | 1990 |
| 20 | 57 | 1121 | 141 | 874 |

Dari tabel di atas dapat diperoleh *ciphertext-ciphertext* yang akan dikirimkan adalah sebagai berikut.

(955,140) (1551,2084) (930,1264) (432,363)

(1311,559) (1732,290) (2183,243) (1345,128)

(2021,551) (609,459) (51,842) (754,1352)

(2008,894) (803,2041) (1616,548) (2281,432)

(2053,2060) (2168,961) (1971,1990) (141,874)

*Ciphertext* itulah yang nantinya akan diterima oleh Rizal. Kelebihan dari penggunaan algoritma Elgamal ini adalah *ciphertext* yang dihasilkan berbeda-beda, meskipun huruf aslinya adalah sama, hal ini dikarenakan karena pemilihan bilangan *k* yang acak.

**3. Proses Dekripsi**

Proses selanjutnya adalah dekripsi. Setelah memperoleh *ciphertext*, maka penerima pesan akan mengubah *ciphertext* $(B_i, \beta_i)$ menjadi *plaintext*,

sehingga dapat dengan mudah membaca isi dari pesan tersebut. Untuk mendekripsi pesan, penerima pesan membutuhkan kunci privat *a*.

Misalkan diberikan suatu kunci publik (*p, g,y*) , dan kunci privat *a*, serta *ciphertext* $(B, \beta)$ ,maka :

$$m = \beta\ (B^a)^{-1}\ \ mod\ p \quad \text{.....} \qquad (3.4)$$

dengan *m* adalah *plaintext*.

Untuk membuktikan persamaan (3.4) , maka akan diuraikan sebagai berikut. Diketahui suatu kunci publik (*p, g,y*) serta *a* sebagai kunci privat. Diberikan suatu *ciphertext* $(B, \beta)$ pada suatu algoritma ElGamal.Berdasarkan dari persamaan (3.1), (3.2), dan (3.3) akan diperoleh :

$$\begin{aligned} \beta\ (B^a)^{-1} &\equiv\ y^k m((g^k)^a)^{-1}\ \ (mod\ p) \\ &\equiv m\ y^k g^{-ak}\ \ (mod\ p) \\ &\equiv m\ \ (g^a)^k\ g^{-ak}\ \ (mod\ p) \\ &\equiv m\ g^0 (mod\ p) \\ &\equiv m\ mod\ p \end{aligned}$$

Yang berarti bahwa *plaintext* dapat ditemukan kembali dari pasangan *ciphertext* $(B, \beta)$. Dengan catatan bahwa $(B^a)^{-1} = B^{-a} = B^{p-1-a}$ mod *p*, karena $\mathbb{Z}_p^*$ merupakan grup siklik berorder *p*-1, dan lambang " $^{-1}$ " menyatakan inversi modulo.

**Algoritma 3.3. Proses Dekripsi**

Input : *Ciphertext* $(B_i, \beta_i)$, $i$ = 1,2,...,*n*., kunci publik ( $p, g, y$) dan kunci privat *a.*

Output : Pesan asli.

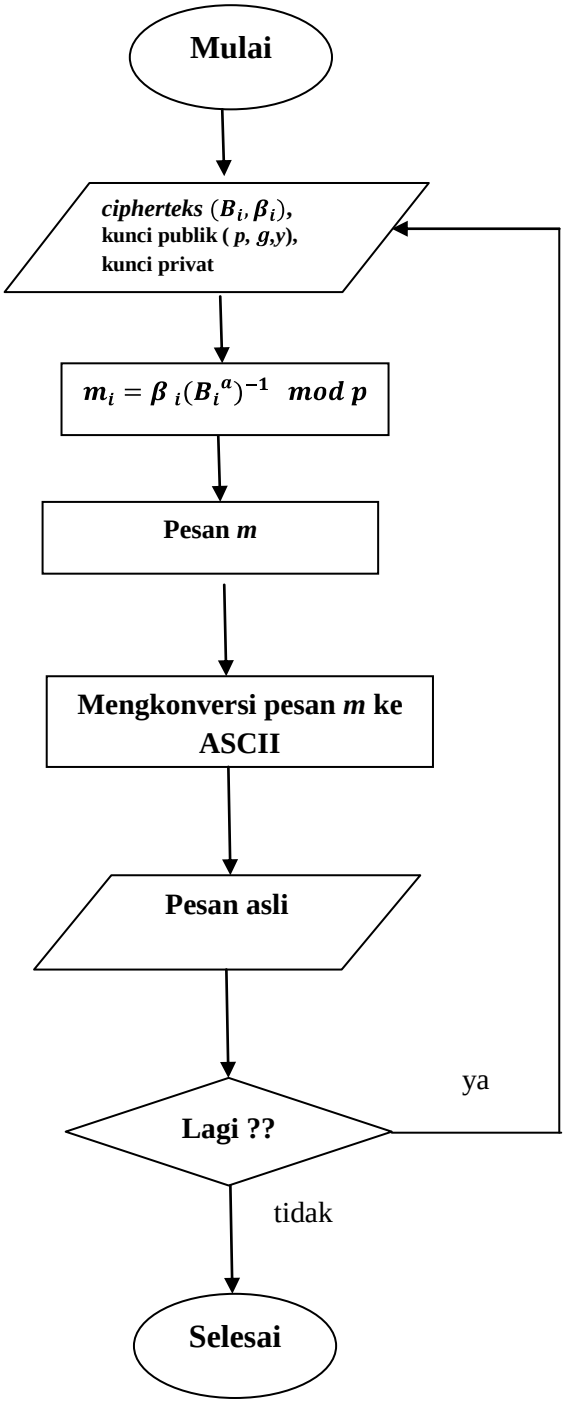
Langkah :

1. Untuk *i* dari 1 sampai *n* :

   Menghitung $m_i = \beta_i (B_i^a)^{-1} \mod p$

2. Akan diperoleh *plaintext* $m_1, m_2, m_{2,...}, m_n$ .
3. Mengkonversikan *plaintext* yang dihasilkan ke dalam kode ASCII, lalu digabungkan kembali.

Di bawah ini, diberikan suatu diagram alir tentang proses dekripsi dari suatu proses pengamanan pesan yang menggunakan sistem kriptografi ElGamal.




**Gambar 3.3. Diagram Alir Proses Dekripsi Suatu pesan**

Contoh 3.3:

Berdasarkan pada contoh sebelumnya. Setelah Rizal menerima *ciphertext* yang telah dikirimkan oleh Ilham. Rizal harus melakukan proses dekripsi agar dapat membaca isi dari pesan tersebut,. Adapun *ciphertext* yang Rizal terima adalah sebagai berikut.

| | | | |
|---|---|---|---|
| (955,140) | (1551,2084) | (930,1264) | (432,363) |
| (1311,559) | (1732,290) | (2183,243) | (1345,128) |
| (2021,551) | (609,459) | (51,842) | (754,1352) |
| (2008,894) | (803,2041) | (1616,548) | (2281,432) |
| (2053,2060) | (2168,961) | (1971,1990) | (141,874) |

Diperoleh kunci publik ($p$,$g$,$y$) = (2357, 2, 1185) dan kunci privat $a$ = 1751. Rizal melakukan proses dekripsi dengan menggunakan persamaan (3.4) . Sebagai hasil perhitungannya, akan disajikan dalam tabel berikut.

**Tabel 3.3. Proses Dekripsi**

| $i$ | $B_i$ | $\beta_i$ | $m_i = \beta_i (B_i)^{605} \mod p$ | Karakter |
|---|---|---|---|---|
| 1 | 955 | 140 | 80 | P |
| 2 | 1551 | 2084 | 97 | a |
| 3 | 930 | 1264 | 115 | s |
| 4 | 432 | 363 | 115 | s |
| 5 | 1311 | 559 | 119 | w |
| 6 | 1732 | 290 | 111 | o |
| 7 | 2183 | 243 | 114 | r |
| 8 | 1345 | 128 | 100 | d |
| 9 | 2021 | 551 | 32 | space |
| 10 | 609 | 459 | 102 | f |
| 11 | 51 | 842 | 105 | i |

| | | | | |
|---|---|---|---|---|
| 12 | 754 | 1352 | 108 | l |
| 13 | 2008 | 894 | 101 | e |
| 14 | 803 | 2041 | 61 | = |
| 15 | 1616 | 548 | 50 | 2 |
| 16 | 2281 | 432 | 50 | 2 |
| 17 | 2053 | 2060 | 48 | 0 |
| 18 | 2168 | 961 | 52 | 4 |
| 19 | 1971 | 1990 | 48 | 0 |
| 20 | 141 | 874 | 57 | 9 |

Setelah melakukan proses dekripsi itu, Rizal mengetahui isi dari pesan yang dikirimkan Ilham, yang berbunyi "Password file=220409", dan tak lain ialah password milik Ilham.

## C. Fungsi *Hash* Satu Arah

Dalam kriptografi, terdapat sebuah fungsi yang digunakan untuk aplikasi keamanan, seperti otentifikasi dan integritas pesan. Fungsi tersebut ialah fungsi *hash*. Fungsi *Hash* adalah fungsi yang menerima masukan string yang panjangnya sembarang dan menkonversikannya menjadi string keluaran yang panjangnya tetap (Rinaldi,2006 : 217). Fungsi *hash* bisa menerima inputan string apa saja. Jika string menyatakan pesan (*message*), maka sembarang pesan *M* yang ukurannya bebas, dimampatkan dengan fungsi *hash* melalui persamaan berikut.

$$MD = H(M) \quad \text{..... (3.5)}$$

dengan *MD* adalah nilai *hash* atau *message dige*st dari fungsi *hash H* dengan masukan pesan *M* .

Ada beberapa cara dalam perhitungan suatu *message digest*. Penulis menggunakan operasi aritmatika yang dapat dikerjakan, misalnya menjumlahkan semua nilai huruf pada pesan, yang sebelumnya pesan sudah dikonversi ke dalam kode ASCII. Kemudian dikenakan operasi modulo 256 pada jumlahan tersebut. Kemudian menambahkan 1 pada nilainya. Adapun algoritmanya adalah sebagai berikut.

**Algoritma 3.4 . Menghitung *Message digest***

*Input* : Pesan yang akan dikirimkan

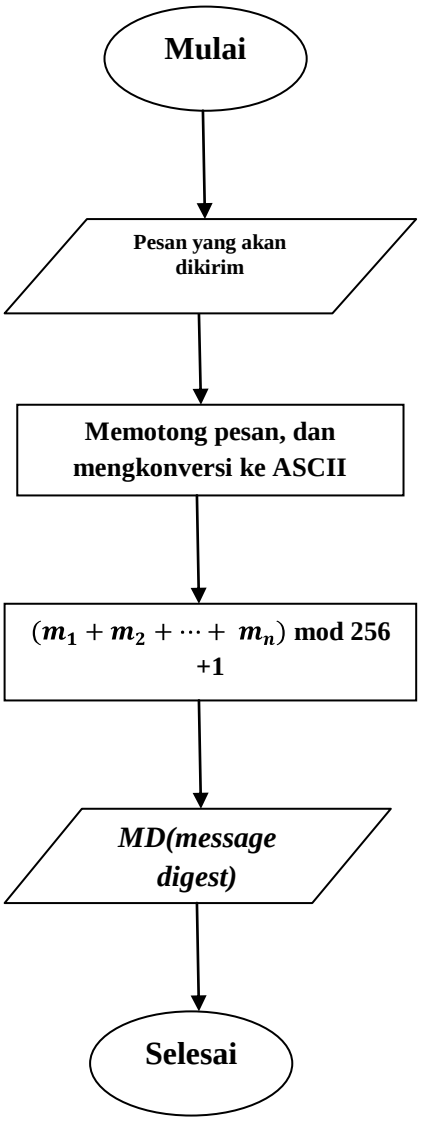
*Output* : Nilai *hash* ( *MD* ).

*Langkah* :

1. Memotong pesan *m* menjadi blok-blok pesan, sehingga satu blok adalah satu karakter pesan.
2. Mengkonversikan masing-masing karakter yang telah diperoleh ke dalam kode ASCII, sehingga diperoleh *plaintext* sebanyak *n* bilangan, yaitu $m_1, m_2, \ldots, m_n$ .
3. Menjumlahkan *plaintext* yang telah dikonversi ke dalam kode ASCII. Lalu melakukan perhitungan

   $$MD = [(m_1 + m_2 + \cdots + m_n) \bmod 256 ] + 1 \quad \text{..... (3.6)}$$

Berdasarkan algoritma di atas, akan diberikan suatu bagan tentang diagram alir proses perhitungan *message digest*, yaitu seperti di bawah ini.




**Gambar 3.4. Diagram Alir Proses Perhitungan *Message Digest***

Contoh 3.4.

Terdapat pesan singkat yang berisi :

Saya pulang sore

Berdasarkan pesan tersebut, akan dicari nilai *MD* (*Message digest*) nya. Langkah pertama yaitu, memecah pesan menjadi beberapa blok . Lalu masing-masing blok dikonversikan ke dalam kode ASCII. Untuk hasil konversinya, dapat dilihat pada tabel berikut.

**Tabel 3.4. Konversi Karakter Pesan Ke Kode ASCII**

| ***i*** | $m_i$ | **karakter** | **ASCII** |
|---|---|---|---|
| 1 | $m_1$ | S | 83 |
| 2 | $m_2$ | a | 97 |
| 3 | $m_3$ | y | 121 |
| 4 | $m_4$ | a | 97 |
| 5 | $m_5$ | <spasi> | 32 |
| 6 | $m_6$ | p | 112 |
| 7 | $m_7$ | u | 117 |
| 8 | $m_8$ | l | 108 |
| 9 | $m_9$ | a | 97 |
| 10 | $m_{10}$ | n | 110 |
| 11 | $m_{11}$ | g | 103 |
| 12 | $m_{12}$ | <spasi> | 32 |
| 13 | $m_{13}$ | s | 115 |
| 14 | $m_{14}$ | o | 111 |
| 15 | $m_{15}$ | r | 114 |
| 16 | $m_{16}$ | e | 101 |

Melihat Tabel 3.4, dapat diketahui bahwa banyaknya karakter $n$ = 16. Lalu menjumlahkan semua karakter yang sudah dikonversi ke dalam kode ASCII menggunakan persamaan (3.6).

$$MD = [(m_1 + m_2 + \cdots + m_{16}) \bmod 256] + 1$$

$$= [(83 + 97 + \cdots + 101) \bmod 256] + 1$$

$$= [1550 \bmod 256] + 1 = 15$$

Jadi nilai *hash* dari pesan singkat tersebut adalah 15.

Perhitungan fungsi *hash* pada skripsi ini hanyalah menjumlahkan karakter-karakter yang telah dikonversi ke dalam kode ASCII yang kemudian hasilnya digunakan dalam proses penandatanganan. Namun, terlepas dari kelebihannya yang bisa memampatkan pesan yang terdiri dari karakter yang banyak, fungsi *hash* ini memiliki kelemahan. Adapun kelemahannya adalah adanya tumbukan (*collision*) dari isi pesan. Maksudnya ada dua buah nilai *hash* yang sama namun berasal dari pesan yang berbeda. Dengan begitu, jika terdapat pihak yang ingin mengubah isi pesan, maka ia akan megacak isi pesan tersebut asalkan nilai *hash* yang dihasilkan sama. Berkembangnya teknologi telah memberikan solusi dari masalah *collision* ini. Ada beberapa cara untuk mengatasi *collision* pada fungsi *hash*, namun dalam skripsi ini, penulis tidak menjelaskan tentang cara mengatasi *collision* pada fungsi *hash*.

Contoh 3.5 ( Rinaldi, 2006 : 229).

> Pada bulan Oktober 2004 ini, suhu udara kota Bandung terasa lebih panas dari hari-hari biasanya. Menurut laporan Dinas Meteorologi Kota Bandung, suhu tertinggi kota Bandung adalah 33 derajat Celcius pada Hari Rabu, 17 Oktober yang lalu. Suhu tersebut sudah menyamai suhu kota Jakarta pada hari-hari biasa. Menurut Kepala Dinas Meteorologi, peningkatan suhu tersebut terjadi karena posisi bumi sekarang ini lebih dekat ke matahari daripada hari-hari biasa. Sebutan Bandung sebagai kota sejuk dan dingin mungkin tidak lama lagi akan tinggal kenangan. Disamping karena faktor alam, jumlah penduduk yang padat, polusi dari pabrik di sekitar Bandung, asap knalpot kendaraan, ikut menambah kenaikan suhu udara kota.

Dari pesan pada contoh di atas, akan dicari nilai hashnya. Untuk ,mencari nilai *hash*nya , dengan menggunakan algoritma mencari *message digest*, maka akan diperoleh suatu nilai hash (*MD*) = 198.

## D. Tanda Tangan Digital Menggunakan Algoritma ElGamal

Yang dimaksud tanda tangan digital disini bukanlah tanda tangan yang di-digitalisasi menggunakan alat *scanner*, namun suatu nilai kriptografis yang bergantung pada pesan dan pengirim pesan. Hal ini kontras dengan tanda tangan pada dokumen biasa, yang hanya bergantung pada pengirim, dan selalu sama untuk semua dokumen. Dengan tangan digital, maka integritas data dapat dijamin, disamping itu dapat digunakan untuk membuktikan keabsahan pengirim, dan nirpenyangkalan.

Tanda tangan digital merupakan suatu mekanisme yang memungkinkan pembuat pesan menambahkan sebuah kode-kode yang bertindak sebagai tanda tangannya. Jadi, tanda tangan digital dapat menjamin integritas dan sumber dari sebuah pesan.

Adapun mekanisme yang digunakan dalam pembuatan tanda tangan digital ialah dengan menggabungkan suatu algoritma publik dengan fungsi *hash*. Prinsip kerja dari tanda tangan digital adalah sebagai berikut. Misalkan A akan membubuhkan tanda tangan pada suatu pesan *m*. Dia menggunakan suatu kunci privat *s*, dan akan diperoleh tanda tangan (*R,T*). B dapat melakukan verifikasi bahwa (*R,T*) merupakan benar tanda tangan dari pesan *m* dengan menggunakan kunci publik . Jadi setelah mengirimkan pesan

*m*, A akan menambahkan nilai *hash* dari suatu pesan yang dikirimkannya. Nilai *hash* tersebut akan digunakan A dalam proses penandatanganan dengan menggunakan kunci privat yang dimiliki oleh A. Setelah mendapatkan pesan dari A, maka B akan mencari nilai *hash* , kemudian melakukan verifikasi dengan menggunakan nilai *hash* dan kunci publik yang dibuat oleh A.

Proses pembuatan tanda tangan digital ElGamal hampir sama dengan sistem kriptografi ElGamal. Yakni terdapat 3 proses, yaitu proses pembuatan kunci , proses penandatanganan, dan proses verifikasi. Berikut akan disajikan gambaran singkat tentang algoritma tanda tangan digital ElGamal.

<table>
<tr><th colspan="2">Parameter buatan yang bersifat publik</th></tr>
<tr><td colspan="2">Seorang pihak yang dapat dipercaya memilih dan kemudian mempublikasikan sebuah bilangan prima besar $p$ dan akar primitif $g$ modulo $p$</td></tr>
<tr><th>Pengirim ( Signer)</th><th>Penerima (verifier)</th></tr>
<tr><th colspan="2">Pembuatan Kunci</th></tr>
<tr><td>Memilih kunci privat $s$, $1 \leq s \leq p-1$<br>Menghitung $v = g^s mod\ p$<br>Mempublikasikan kunci publik ($p$, $g$,$v$)</td><td></td></tr>
<tr><th colspan="2">Proses Penandatanganan</th></tr>
<tr><td>Menghitung MD dari pesan<br>Memilih $e$, yang relatif prima dengan $p$ -1<br>Menghitung :<br>$R = g^e mod\ p$ dan<br>$T = (MD - sR)e^{-1} mod\ (p-1)$</td><td></td></tr>
</table>

| Proses Verifikasi | |
|---|---|
| | Mengecek bahwa $1 \leq R \leq p-1$ terpenuhi<br>Menghitung $v^R R^T \bmod p$, kemudian diperiksa bahwa $v^R R^T \equiv g^{MD} \bmod p$ |

**Gambar 3.5. Algoritma Tanda Tangan Digital ElGamal (Hoffstein, Jill, and Silverman, 2008 :443)**

**1. Proses Pembentukan Kunci**

Proses pertama dalam pembuatan tanda tangan digital adalah proses pembuatan kunci. Proses ini dilakukan oleh pihak pengirim pesan, yang nantinya akan diperoleh kunci privat dan kunci publik. Pada proses ini dibutuhkan sebuah bilangan prima $p$ yang digunakan untuk membentuk grup $\mathbb{Z}_p^*$, elemen primitif $g$ dan sebarang $s \in \{1,2,3,\dots,p-1\}$. Diketahui bahwa $s$ merupakan kunci privat yang akan dipegang oleh pengirim pesan. Lalu mencari kunci publik dengan cara menghitung

$$v = g^s mod\ p \qquad (3.7)$$

Diperoleh ($p$,$g$,$v$) adalah kunci publik, yang akan diberitahukan kepada penerima pesan.

Algoritma 3.5. Pembentukan Kunci

*Input* : bilangan prima $p > 255$ ,dan elemen primitif $g$ mod $p$ , dan $s$.

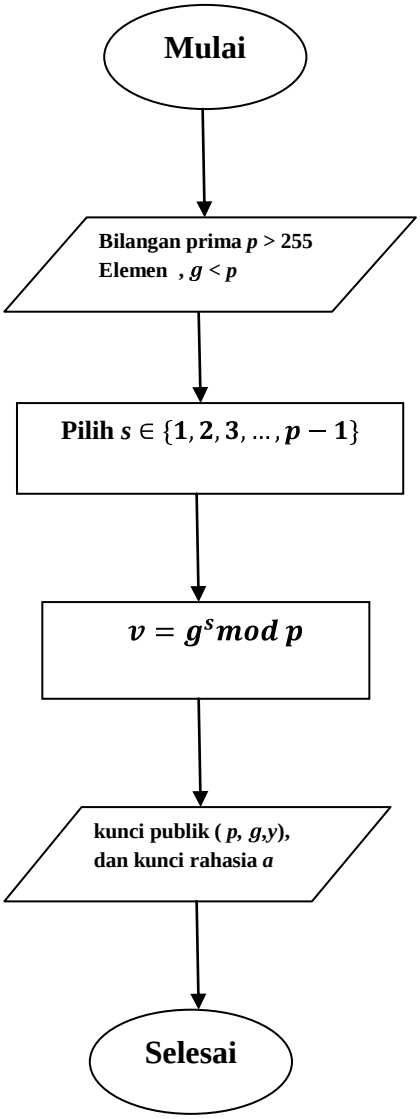
*Output* : kunci publik ($p$, $g$,$v$)

*Langkah* :

1. Pilih bilangan prima $p$.

2. Pilih dua buah bilangan acak $g$ dan *s*, dimana $s \in \{1,2,3,\dots,p-1\}$
3. Hitung $v = g^s mod\ p$
4. Publikasikan nilai *p*, $g$, dan *v*. Namun nilai *s* dirahasiakan.

Berdasar pada algoritma pembentukan kunci di atas, dapat dibuat suatu diagram alir tentang pembuatan kunci tanda tangan digital. Adapun bagannya seperti di bawah ini.




**Gambar 3.6. Diagram Alir Pembentukan Kunci Tanda Tangan Digital**

Contoh 3.6:

Ilham ingin membubuhkan suatu tanda tangan pada pesan yang akan dikirimkannya kepada Rizal. Langkah awalnya, Ilham akan membangkitkan kunci terlebih dahulu. Ilham memilih bilangan prima $p$ = 21739 , dan $g$ = 7. Ia juga memilih *s* = 15140. Ilham akan memperoleh $v = 7^{15140}\ mod\ 21739 = 17702$ dengan menggunakan persamaan 3.7. Dengan begitu, kunci privat *s* adalah 15140 , dan kunci publik yang diberikan adalah (*p*,*g*,*v*) = (21739,7,17702). Kunci publik tersebut akan digunakan untuk memverifikasi tanda tangan digital yang dibubuhkan pada pesan.

**2. Proses Penandatanganan (*Signing*)**

Setelah proses pembentukan kunci, maka proses selanjutnya adalah proses penandatanganan (*signing*). Pada proses ini, dibutuhkan kunci privat dan juga nilai *hash* (*MD*) dari suatu pesan *m*. Sebelum memberi tanda tangan pada dokumen, sang pengirim terlebih dahulu menghitung nilai *hash* dari pesan yang akan dikirimkan, dimana dengan menggunakan algoritma yang telah dijelaskan pada subbab sebelumnya. Nilai *hash* yang dihasilkan adalah $MD \in \{1,2,3,\dots,p-2\}$. Kemudian memilih suatu bilangan acak *e* yang berada dalam $\{1,2,3,\dots,p-2\}$, dan *e* saling prima dengan *p*-1, dengan kata lain *gcd*(*e*,*p-1*) = 1. Selanjutnya pembuat tanda tangan (*signer*) akan melakukan perhitungan berikut.

$$R = g^e mod\ p \qquad (3.8)$$

dan

$$T = (MD - sR)e^{-1} mod\ (p-1) \text{.....(3.9)}$$

$e^{-1}$ merupakan invers dari *e* mod (*p-1*). Maka tanda tangan pada dokumen tersebut adalah pasangan dari (*R,T*).

**3. Proses Verifikasi**

Setelah pesan telah sampai kepada pihak penerima, maka penerima akan melakukan proses verifikasi. Untuk melakukan proses ini, penerima pesan menggunakan kunci publik (*p,g,v*) yang telah diberikan dari pengirim pesan. Si penerima memperoleh pesan yang berupa dokumen yang telah dibubuhi tanda tangan digital. Dalam hal ini, dokumen bisa saja berupa *plaintext* maupun *ciphertext*. Tergantung dari perjanjian antar pihak yang bersangkutan. Sebelumnya, akan dihitung terlebih dahulu nilai *hash* dari dokumen yang diterima. Kemudian penerima akan memverifikasi tanda tangan (*R,T*). Terlebih dahulu ia akan mengecek apakah memenuhi *R* berada dalam $\{1,2,3,\dots,p-1\}$. Setelah memenuhi, dilanjutkan dengan mengecek apakah memenuhi persamaan $v^R R^T \equiv g^{MD} \bmod p$. Jadi verifikasi dikatakan berhasil jika memenuhi 2 keadaan berikut :

1. $1 \leq R \leq p-1$
2. Memenuhi $v^R R^T \equiv g^{MD} \bmod p$ ..... (3.10)

Lalu akan ditunjukkan bahwa proses verifikasi telah bekerja. Jika *T* dihitung berdasarkan (3.9), maka

$$v^R R^T \equiv g^{sR} g^{e(MD-sR)e^{-1}} \equiv g^{MD} \bmod p \text{.....} \quad (3.11)$$

Sebaliknya jika (3.10) terpenuhi untuk pasangan tanda tangan (*R*,*T*) dan jika *e* adalah logaritma diskret dari *R* dengan basis *g*, atau bisa ditulis dengan $R = g^e$, maka

$$v^R R^T \equiv g^{sR} g^{eT} \equiv g^{sR+eT} \equiv g^{MD} \bmod p.$$

$g$ adalah elemen primitif modulo *p*, mengakibatkan

$$sR + eT \equiv MD \bmod (p\text{-}1)$$

Jika *e* dan *p-1* adalah saling prima, maka akan mengakibatkan (3.9).

Contoh 3.7 (Buchmann, 2000 : 244)

Diberikan bilangan prima $p$ = 23, $g$ = 7, dan *s*=6. Selanjutnya akan dicari kunci publiknya dengan menghitung $v = g^s mod\ p$= $7^6$ mod 23 = $7^2.7^2.7^2$ mod 23 = 27 mod 23 = 4. Jadi diperoleh kunci publik (*p*,*g*,*v*) = (23,7,4). Kunci privatnya adalah $s$ = 6. Misalkan suatu pesan *m* akan diberi tanda tangan. Nilai *hash* dari *m* adalah *MD* = 7. Dipilih e = 5, sehingga diperoleh $R = g^e mod\ p$ = 17. Nilai dari $e^{-1} mod\ (p-1)$=9. Selanjutnya menghitung $T = (MD - sR)e^{-1} mod\ (p-1)$ = (7-6.17)9 mod 22 = 3 . Jadi tanda tangannya adalah (*R*,*T*) = (17, 3). Untuk melakukan verifikasi tanda tangan tersebut digunakan kunci publik. Lalu menghitung $v^R R^T mod\ p =$ $4^{17}17^3$mod 23 = 5. Selain itu, juga menghitung $g^{MD}$ mod $p$ = $7^7$mod 23 = 5. Perhitungan tersebut telah memenuhi persamaan (3.10) maka tanda tangan berhasil diverifikasi.

Contoh 3.8 :

Ilham ingin mengirimkan suatu pesan dengan tanda tangan kepada Rizal. Adapun isi dari pesan tersebut adalah "Password file=220409 ". Langkah awal yang harus Ilham lakukan adalah membuat kunci dan mencari nilai *hash* dari pesan tersebut. Berdasarkan Contoh 3.6 telah diperoleh kunci privat *s* adalah 15140 , dan kunci publik yang diberikan adalah (*p*,*g*,*v*) = (21739,7,17702). Lalu selanjutnya Ilham mencari nilai hashnya, yaitu diperoleh *MD* = 130. Setelah mendapatkan kunci, maka dilanjutkan dengan proses penandatanganan. Ilham memilih bilangan *e* yang saling prima dengan *p-1*, dipiih *e* = 10727. Dan invers dari *e* modulo *p-1* adalah 6353. Lalu dihitung $R = g^e mod\ p$ = 15775.Selanjutnya menghitung $T = (MD - sR)\ e^{-1}$ mod (*p-1*) = (130 – 15140.15775).6353= 598. Jadi tanda tangannya adalah (*R*,*T*) = (15775, 5802). Ilham mengirimkan pesan beserta tanda tangan kepada Rizal. Selanjutnya Rizal akan memverifikasi tanda tangan tersebut dengan kunci publik yang dia miliki. Berdasarkan perhitungan tanda tangan tersebut, Rizal menghitung $v^R R^T mod\ p = 17702^{15775} 15775^{598}$ mod 21739 = 11045, juga menghitung $g^{MD}$ mod $p$ = $7^{130}$ mod 21739 = 11045. Perhitungan yang dilakukan oleh Rizal memenuhi persamaan (3.10) maka tanda tangan terverifikasi.

**Contoh 3.10** .

Misalkan Bapak Gunawan dan Bapak Masfuri, merupakan dosen di suatu universitas, akan menyampaikan suatu dokumen penting kepada bagian penyerahan nilai mahasiswa. Beliau menitipkan kepada salah seorang

mahasiswa mereka untuk disampaikan pada bagian penilaian mahasiswa. Dikarenakan isi dokumen penting, maka kedua dosen tersebut memberikan tanda tangan pada dokumen tersebut. Berikut isi dari dokumen tersebut.

| |
|---|
| Dengan ini, diberitahukan bahwa mahasiswa kami yang bernama Wahyu Nur Habibi dengan NIM 07305141001 telah menyelesaikan ujian TA, dan mendapatkan nilai 86,5 dengan indeks A. Demikian telah dilakukan penilaian secara menyeluruh.<br><br>*Ttd* *Ttd*<br><br>**Gunawan** **Masfuri** |

Dokumen tersebut telah ditandangani oleh dua orang, ini berarti terdapat dua kunci publik yang akan diberikan kepada pihak penilaian mahasiswa. Di bawah ini adalah proses pembentukan kunci oleh Bapak Gunawan dan Bapak Masfuri.

Berdasarkan dokumen di atas, untuk menghitung nilai *hash*nya, inputnya adalah isi dokumen tersebut, yaitu di bawah ini.

| |
|---|
| Dengan ini, diberitahukan bahwa mahasiswa kami yang bernama Wahyu Nur Habibi dengan NIM 07305141001 telah menyelesaikan ujian TA, dan mendapatkan nilai 86,5 dengan indeks A. Demikian telah dilakukan penilaian secara menyeluruh. |

**Pihak I (Bapak Gunawan)**

Bapak Gunawan memilih bilangan besar $p_1$ = 15137 , dan $g_1$ = 3. Serta ia memilih $s_1$ = 14121. Bapak Gunawan melakukan perhitungan dengan

menggunakan persamaan 3.7, kemudian diperoleh $v_1 = 3^{14121} mod\ 15137$ .= 15011. Kunci privat $s_1$ adalah 14121 , dan kunci publik yang diberikan adalah ($p_1$, $g_1$,$v_1$) = (15137, 3, 15011). Berdasar dari isi dokumen tersebut diperoleh nilai *hash*nya (*MD*) yaitu 111. Dan dipilih nilai $e_1$ = 13217. Selanjutnya adalah proses pembuatan tanda tangan. Pertama bapak Gunawan melakukan perhitungan $R_1 = {g_1}^{e_1} mod\ p_1$ = $3^{13217}$mod 15137 = 5003. Kemudian diperoleh $T_1 = (MD - s_1 R_1)\ {e_1}^{-1}$ mod ($p_1$-1) = (111-14121.5003)$13217^{-1}$ mod 15136 = 12332. Jadi tanda tangan milik Bapak Gunawan adalah ($R_1, T_1$) = ( 5003,12332).

**Pihak II (Bapak Masfuri)**

Dan Bapak Masfuri memilih bilangan besar $p_2$ = 17011 , dan $g_2$ = 2. Serta ia memilih $s_2$ = 16982. Dengan menggunakan persamaan 3.7 Bapak Masfuri akan memperoleh $v_2 = 2^{16982} mod\ 17011 = 4688$ . Dengan begitu, kunci privat $s_2$ adalah 16982 , dan kunci publik yang diberikan adalah ($p_2$, $g_2$,$v_2$) = (17011,2,4688). Berdasar dari isi dokumen tersebut diperoleh nilai *hash*nya yaitu 111. Dan dipilih nilai $e_2$ = 13313 . Pertama bapak Masfuri melakukan perhitungan $R_2 = {g_2}^{e_2} mod\ p_2$ = $2^{13313}$mod 17011 = 8603. Kemudian diperoleh $T_2 = (MD - s_2 R_2)\ {e_2}^{-1}$ mod ($p_2$-1) = (111- 16982.8603)$13313^{-1}$ mod 17010 = 16885. Jadi tanda tangan milik Bapak Masfuri adalah ($R_2, T_2$) = ( 8603,16885).

**Verifikasi oleh Pihak III (Bagian Penilaian Mahasiswa)**

Nilai *hash(MD)* dari dokumen = 111

1. Tanda tangan I ( Bp. Gunawan)

   Diperoleh $(R_1, T_1)$ = ( 5003,12332).

   Kunci publik = ($p_1$, $g_1$,$v_1$) = (15137,3,15011)

   Lalu menghitung $v_1{}^{R_1} R_1{}^{T_1} mod\ p = 15011^{5003} 5003^{12332}$ mod 15137 = 5783. Pihak III juga menghitung $g_1{}^{MD}$ mod $p_1$ = $3^{111}$ mod 15137 = 5783. Karena $v_1{}^{R_1} R_1{}^{T_1} \equiv g_1{}^{MD}$ mod $p_1$ , maka verifikasi telah dilakukan.

2. Tanda tangan II (Bp. Masfuri)

   Diperoleh $(R_2, T_2)$ = ( 8603,16885).

   Kunci publik = ($p_2$,$g_2$,$v_2$) = (17011,2,4688)

   Lalu menghitung $v_2{}^{R_2} R_2{}^{T_2} mod\ p_2 = 4688^{8603} 8603^{16885}$mod 17011 = 12240. Pihak III juga menghitung $g_2{}^{MD}$ mod $p_2$ = $2^{111}$ mod 17011 =12240. Karena $v_2{}^{R_2} R_2{}^{T_2} \equiv g_2{}^{MD}$ mod $p_2$ , maka verifikasi telah dilakukan.

Karena saat proses verifikasi cocok, maka dapat dikatakan bahwa dokumen yang akan diberikan kepada bagian penilaian mahasiswa tersebut sah, berasal dari Bapak Gunawan dan Bapak Masfuri, tanpa ada pengubahan isi dokumen dari pihak lain.

**Contoh 3.11**

Berdasarkan Contoh 3.10 , seorang mahasiswa yang diberi kepercayaan untuk mengantarkan dokumen tersebut, ternyata mengubah isi dari dokumen

tersebut. Adapun mahasiswa tersebut hanya mengubah bagian dari isi surat saja. Di bawah ini adalah surat yang telah diubah oleh mahasiswa tersebut.

> Dengan ini, diberitahukan bahwa mahasiswa kami yang bernama Dimas Setya Aji dengan NIM 07305141091 telah menyelesaikan ujian TA, dan mendapatkan nilai 86,5 dengan indeks A. Demikian telah dilakukan penilaian secara menyeluruh.

Setelah memperoleh dokumen dari mahasiswa tersebut, bagian Penilaian Mahasiswa akan memverifikasi tanda tangan pada milik Bapak Gunawan = ($R_1$, $T_1$) = ( 5003, 12332) dan Bapak Masfuri = ($R_2$, $T_2$) = ( 8603, 16885).

**Verifikasi oleh Pihak III (Bagian Penilaian Mahasiswa)**

Nilai *hash(MD)* dari dokumen = 254

1. Tanda tangan I ( Bp. Gunawan)

   Diperoleh ($R_1, T_1$) = ( 5003, 12332).

   Kunci publik = ($p_1$, $g_1$ ,$v_1$) = (15137, 3, 15011)

   Lalu menghitung $v_1{}^{R_1} R_1{}^{T_1} mod\ p_1 = 15011^{5003} 5003^{12332}$mod 15137 = 5783. Selanjutnya juga dihitung $g_1{}^{MD}$ mod $p_1$ = $3^{254}$ mod 15137 = 14150. Karena $v_1{}^{R_1} R_1{}^{T_1} \not\equiv g_1{}^{MD}$ mod $p_1$ , verifikasi tanda tangan tidak cocok.

2. Tanda tangan II (Bp. Masfuri)

   Diperoleh ($R_2, T_2$) = ( 8603, 16885).

   Kunci publik = ($p_2$, $g_2$,$v_2$) = (17011, 2, 4688)

   Lalu menghitung $v_2{}^{R_2} R_2{}^{T_2} mod\ p_2 = 4688^{8603} 8603^{16885}$mod 17011 = 12240. Selanjutnya dihitung $g_2{}^{MD}$ mod $p_2$ = $2^{254}$ mod 17011 = 4128.

Karena ${v_2}^{R_2}{R_2}^{T_2} \not\equiv {g_2}^{MD} \bmod p_2$ , maka verifikasi tanda tangan tidak cocok.

Karena saat proses verifikasi tidak cocok, maka dapat dikatakan bahwa dokumen yang akan diberikan kepada bagian penilaian mahasiswa tersebut tidak sah, berasal dari Bapak Gunawan dan Bapak Masfuri, dan diindikasikan telah terjadi pengubahan isi dokumen yang dikirimkan.

# BAB IV

# PENUTUP

## A. Kesimpulan

Kesimpulan yang dapat diambil dalam skripsi ini sebagai berikut :

Penerapan sistem kriptografi ElGamal pada tanda tangan digital terdapat pada proses pembentukan kunci, penandatanganan, serta verifikasi. Perhitungannya ketiga proses tersebut berdasar pada masalah logaritma diskret pada grup $\mathbb{Z}_p^*$. Proses pembuatan tanda tangan digital pada suatu dokumen dengan menggunakan algoritma ElGamal adalah sebagai berikut:

1. Pembentukan kunci yang dilakukan oleh pengirim pesan. Langkah-langkah pembentukan kunci meliputi:
   a) Pilih bilangan prima $p$.
   b) Pilih dua buah bilangan acak $g$ dan $s$, dimana $s \in \{1,2,3,\dots,p-1\}$
   c) Hitung $v = g^s mod\ p$
   d) Publikasikan nilai $p$, $g$,dan $v$. Namun nilai $s$ dirahasiakan.

   Dari proses pembentukan kunci, diperoleh kunci publik ($p$, $g$, $v$) dan kunci $a$.
2. Proses Penandatanganan. Proses penandatanganan dilakukan oleh pengirim pesan. Adapun langkah-langkah penandatanganan suatu dokumen adalah sebagai berikut.
   a) Menghitung nilai *hash* (*MD*) suatu dokumen dengan langkah sebagai berikut.

1) Memotong pesan *m* menjadi blok-blok pesan, sehingga satu blok adalah satu karakter pesan.
2) Mengkonversikan masing-masing karakter yang telah diperoleh ke dalam kode ASCII, sehingga diperoleh *plaintext* sebanyak *n* bilangan, yaitu $m_{1,}\ m_{2\ ,\ldots,}\ m_n$ .
3) Menjumlahkan *plaintext* yang telah dikonversi ke dalam kode ASCII. Lalu melakukan perhitungan berikut :

   $MD\ =\ [(m_1 + m_2 + \cdots +\ m_n)$ mod 256] + 1.

b) Menghitung nilai kriptografis suatu dokumen. Langkah – langkah dalam menghitung nilai kriptografis adalah sebagai berikut.
1) Memilih *e,* yang relatif prima dengan *p*-1
2) Menghitung :

   $R = g^e mod\ p$ dan $T = (MD - sR)e^{-1} mod\ (p - 1)$.

3) Membubuhkan tanda tangan (*R, T*) pada dokumen.

(*R, T*) inilah yang dinamakan nilai kriptografis (tanda tangan digital).

c) Mengirimkan dokumen yang telah dibubuhi dengan tanda tangan digital.

3. Proses verifikasi tanda tangan digital. Pada proses verifikasi ini, dilakukan oleh pihak penerima pesan. Langkah – langkah dalam proses verifikasi tanda tangan adalah sebagai berikut.
a) Menghitung nilai *MD.*
b) Mengecek bahwa $1 \leq R \leq p\text{-}1$ terpenuhi

c) Menghitung $v^R R^T \bmod p$.

d) Dan diperiksa bahwa $v^R R^T \equiv g^{MD} \bmod p$ .

Jika perhitungan $v^R R^T \equiv g^{MD} \bmod p$ terpenuhi, maka dokumen yang dikirimkan dikatakan masih asli atau berasal dari pengirim yang sebenarnya.

**B. Saran**

Setelah membahas proses pembuatan tanda tangan digital menggunakan algoritma ElGamal atas $\mathbb{Z}_p^*$ pada skripsi ini, penulis ingin menyampaikan beberapa saran sebagai berikut :

1. Sistem kriptografi ElGamal merupakan salah satu algoritma yang aman digunakan dalam pengamanan pesan. Meskipun termasuk dalam algoritma yang aman, kunci publik juga harus dijaga keamanannya dengan membuat kunci yang berbeda setiap melakukan komunikasi agar tidak dimanipulasi oleh pihak-pihak yang tidak bertanggungjawab.
2. Dalam proses penandatanganan, pada saat perhitungan nilai *hash* terdapat kelemahan yakni *collision* pada nilai *hash*. Oleh sebab itu diperlukan metode pencarian nilai *hash* yang aman dari *collision*, misalnya dilakukan *double hashing*, *linear probing*, maupun metode yang lainnya. Bagi pembaca yang ingin membahas lebih lanjut tentang tanda tangan digital, bisa dilakukan penelitian tentang penanganan masalah *collision* pada fungsi *hash.*

# DAFTAR PUSTAKA


Buchmann, Johannes A. 2000. *Introduction to Cryptography*. New York : Springer-Verlag.

Fraleigh, John B. 2000. *A First Course in Abstract Algebra*. Sixth Edition. New York : Addison- Wesley Publishing Company.

Hoffstein,Jeffrey , Pipher, Jill, and Silverman, Joseph H. 2008. *An Introduction to Mathematical Cryptography*. New York : Springer-Verlag.

Iskandar,Kusrini , Sismoro,Heri.2004.*Struktur Data dan Pemrograman dengan Pascal*.Yogyakarta :Andi offset.

Mao, Wenbo. 2004. *Modern Cryptography Theory & Practice*. New Jersey : Prentice-Hall Publisher.

Menezes, Oorcshot, and Vanstone. 1996. *Handbook of Applied Cryptography*. Florida : CRC Press.

Mollin,Richard A. 2007. *An Introduction to Cryptography.* 2nd edition .New York : CRC Press.

Munir Rinaldi. 2006. *Kriptografi*. Bandung: Informatika Bandung.

Trappe, Wade , Washington, Lawrence C. 2006. *Introduction to Cryptography with Coding Theory*. 2nd edition. New Jersey : Prentice Hall.

Sangadji. 2002. "Uji Primalitas". *Lokakarya Komputasi dalam Sains dan Teknologi Nuklir (XIII).* (3-4 Juli 2002). Jakarta: PPIN-BATAN.

Stallings, Williams , 2005. *Cryptography and Network SecurityPrinciples and Practices* 4th edition. New Jersey : Pearson

Stinson, D.R. 2006. *Cryptography Theory and Practice*. Florida : CRC Press.

Sukirman. 2006. *Pengantar Teori Bilangan.* Yogyakarta : Hanggar Kreator.

_____. 2005. *Pengantar Aljabar Abstrak*. Yogyakarta : FMIPA UNY Yogyakarta.

_____. 2006. *Aljabar Abstrak Lanjut.* Yogyakarta : Hanggar Kreator.

Wikipedia.(2011).*ElGamalSignatureScheme*/http://en.wikipedia.org/wiki/ ElGamal_ signature_scheme. Tanggal akses 11 januari 2011 pukul 16.23.

## Lampiran 1: Tabel Kode ASCII

### Kode ASCII ( 0 -127 )

| No | Kode |
|---|---|
| 0 | NULL (null) |
| 1 | SOH (start of heading) |
| 2 | STX (start of text) |
| 3 | ETX (end of text) |
| 4 | EOT (end of transmission) |
| 5 | ENQ (enquiry |
| 6 | ACK (acknowledge) |
| 7 | BEL (bell) |
| 8 | BS (backspace) |
| 9 | TAB (horizontal tab) |
| 10 | LF (new line) |
| 11 | VT (vertical tab) |
| 12 | FF (new page) |
| 13 | CR (carriage return) |
| 14 | SO (shift out) |
| 15 | SI (shift in) |
| 16 | DLE (data link escape) |
| 17 | DC1 (device control 1) |
| 18 | DC2 (device control 2) |
| 19 | DC3 (device control 3) |
| 20 | DC4 (device control 4) |
| 21 | NAK (negative acknowledge) |
| 22 | SYN (synchronus idle) |
| 23 | ETB (end of trans. blok) |
| 24 | CAN (cancel) |
| 25 | EM (end of medium) |
| 26 | SUB (substitute) |
| 27 | ESC (escape) |
| 28 | FS (file separator) |
| 29 | GS (group separator) |
| 30 | RS (record separator) |
| 31 | US (unit separator) |
| 32 | Space |
| 33 | ! |
| 34 | " |
| 35 | # |
| 36 | $ |
| 37 | % |
| 38 | & |
| 39 | ' |
| 40 | ( |
| 41 | ) |
| 42 | * |

| No | Kode |
|---|---|
| 43 | + |
| 44 | , |
| 45 | - |
| 46 | . |
| 47 | / |
| 48 | 0 |
| 49 | 1 |
| 50 | 2 |
| 51 | 3 |
| 52 | 4 |
| 53 | 5 |
| 54 | 6 |
| 55 | 7 |
| 56 | 8 |
| 57 | 9 |
| 58 | : |
| 59 | ; |
| 60 | < |
| 61 | = |
| 62 | > |
| 63 | ? |
| 64 | @ |
| 65 | A |
| 66 | B |
| 67 | C |
| 68 | D |
| 69 | E |
| 70 | F |
| 71 | G |
| 72 | H |
| 73 | I |
| 74 | J |
| 75 | K |
| 76 | L |
| 77 | M |
| 78 | N |
| 79 | O |
| 80 | P |
| 81 | Q |
| 82 | R |
| 83 | S |
| 84 | T |
| 85 | U |

| No | Kode |
|---|---|
| 86 | V |
| 87 | W |
| 88 | X |
| 89 | Y |
| 90 | Z |
| 91 | [ |
| 92 | \ |
| 93 | ] |
| 94 | ^ |
| 95 | _ |
| 96 | ` |
| 97 | a |
| 98 | b |
| 99 | c |
| 100 | d |
| 101 | e |
| 102 | f |
| 103 | g |
| 104 | h |
| 105 | i |
| 106 | j |
| 107 | k |
| 108 | l |
| 109 | m |
| 110 | n |
| 111 | o |
| 112 | p |
| 113 | q |
| 114 | r |
| 115 | s |
| 116 | t |
| 117 | u |
| 118 | v |
| 119 | w |
| 120 | x |
| 121 | y |
| 122 | z |
| 123 | { |
| 124 | \| |
| 125 | } |
| 126 | ~ |
| 127 | DEL |

## Lampiran 2

### Program Matlab untuk Menghitung $a^k \bmod p$

```
function y=modulo(g,a,p)
bin=dec2bin(a);
lbin=length(bin);
y=1;
if(a==0)
    y=1;
end
if(a~=0)
    y1=g;
    if(bin(lbin)=='1')
        y=g;
    end
    for i= lbin-1:-1:1
        y1 =mod(y1*y1,p);
        if(bin(i)=='1')
            y=mod(y*y1,p);
        end
    end
end
```

### Program Matlab untuk Menghitung $x^{-1} \bmod p$

```
function invc=DKR(x,p)
x=mod(x,p);
u=[1 0 p];
v=[0 1 x];
ulang=1;
while(ulang==1)
    if(v(3)==0)
        invc=[];
        ulang=0;
        break;
    end
    if(v(3)==1)
        invc=mod(v(2),p);
        ulang=0;
        break
    end
    if(ulang~=0)
        q=floor(u(3)/v(3));
        t=[u(1)-q*v(1) u(2)-q*v(2) u(3)-q*v(3)];
        u=v;
        v=t;
    end
end
```

**Lampiran 3**

## Program Matlab Untuk Pengiriman Pesan Menggunakan Sistem Kriptografi ElGamal

**Contoh 3.1 (Pembangkitan Kunci )**

```
>> p= 2357;g=2;a=1751;
>> KR=modulo(g,a,p)
KR =

      1185

>> KP=[p g KR]
KP =

      2357          2       1185
```

**Contoh 3.2 (Proses Enkripsi)**

**Mengkonversi dari huruf ke kode ASCII**

```
>> message='Password file=220409';

>> convert=double(message)
convert =

   80   97  115  115  119  111  114  100   32  102  105  108  101   61   50
50   48   52   48   57

>> m1=80;
>> k1=141;
>> B=modulo(g,k1,p)
B =

  955
>> Beta1=mod(m1,p)
Beta1 =

   80
```

```
>> Beta2=modulo(kunci,k1,p)
Beta2 =

   591
>> Beta=mod(Beta1*Beta2,p)
Beta =

   140

>> cipherteks=[B Beta]
cipherteks =

   955   140
```

**Contoh 3.3 Proses Dekripsi**

```
>> v=p-1-kunci
v =

   605

>> con1=mod(Beta,p)
con1 =

   140

>> con2=modulo(B,v,p)
con2 =

   674

>> mess=mod(con1*con2,p)
mess =

    80

>> rubah=char(mess)

rubah =

P
```

**Lampiran 4**

**Program Untuk Melakukan Proses Penandatanganan Digital Menggunakan Sistem Kriptografi ElGamal**

**Contoh 3.4**

**Menghitung Nilai Hash (*Message Diggest*)**

```
>> message='Saya pulang sore';
>> tran=double(message);
>> jum=sum(tran)
jum =

      1550

>> hash=mod(jum,256)+1
hash =

   15
```

**Contoh 3.6**

**Proses Pembentukan Kunci Privat dan Kunci Publik**

```
>> p=21739;g=7;s=15140;
>> privat=s
privat =

     15140

>> publik=modulo(g,s,p)
publik =

     17702

>> kunci=[p,g,publik]
kunci =

     21739          7       17702
```

**Contoh 3.8**

**Proses Penandatangan Digital pada Dokumen**

```
>> mess='Password file=220409';
>> dub=double(mess);
```

```
>> juml=sum(dub)
juml =

       1665

>> hash=mod(juml,256)+1
hash =

  130

>> e=10727;
>> q=p-1;
>> gcd(e,p)
ans =

    1

>> T1=(hash-s*R);
>> w1=mod(T1,q)
w1 =

       2036

>> w2=DKR(e,q)
w2 =

       6353

>> T=mod(w1*w2,q)
T =

  598

>> tandatangan=[R T]
tandatangan =

      15775         598
```

**Proses Verifikasi Tanda Tangan**

```
>> v1=modulo(publik,R,p)
v1 =

       8957

>> v2=modulo(R,T,p)
```

```
v2 =
     16675

>> ver1=mod(v1*v2,p)
ver1 =

     11045

>> ver2=modulo(g,hash,p)
ver2 =

     11045
```

**Lampiran 5**

## Program Untuk Melakukan Verifikasi Tanda Tangan Yang Telah Mengalami Pengubahan oleh Pihak Ketiga

**Contoh 3.10**

1. Proses Perhitungan Nilai Hash

```
>> message1='Dengan ini, diberitahukan bahwa mahasiswa kami yang
bernama Wahyu Nur Habibi dengan NIM 07305141001 telah
menyelesaikan ujian TA, dan mendapatkan nilai 86,5 dengan indeks A.
Demikian telah dilakukan penilaian secara menyeluruh.';
>> mess=double(message1);
>> jum1=sum(mess)

jum1 =

     20334

>> hash1=mod(jum1,256)+1

hash1 =

   111
```

2. Proses Pembentukan Kunci

```
>> p1=15137;g1=3;s1=14121;
>> p2=17011;g2=2;s2=16982;
>> v1=modulo(g1,s1,p1)
v1 =

     15011

>> privat1=s1
privat1 =

     14121

>> publik1=[p1 g1 v1]
publik1 =

     15137          3       15011
```

```
>> v2=modulo(g2,s2,p2)
v2 =

       4688

>> privat2=s2
privat2 =

      16982

>> publik2=[p2 g2 v2]
publik2 =

      17011           2        4688
```

3. Proses Penandatanganan

```
>> e1=13217;q1=p1-1;
>> gcd(e1,q1)
ans =

     1

>> e2=13313;q2=p2-1;
>> gcd(e2,q2)
ans =

     1

>> R1=modulo(g1,e1,p1)
R1 =

       5003

>> R2=modulo(g2,e2,p2)
R2 =

       8603

>> md1=(hash1-s1*R1);
>> t1=mod(md1,q1);
>> t2=DKR(e1,q1);
>> T1=mod(t1*t2,q1)
T1 =

      12332
```

```
>> md2=(hash1-s2*R2);
>> t3=mod(md2,q2);
>> t4=DKR(e2,q2);
>> T2=mod(t3*t4,q2)
T2 =

    16885

>> Tandatangan1=[R1 T1]
Tandatangan1 =

    5003    12332

>> Tandatangan2=[R2 T2]
Tandatangan2 =

    8603    16885
```

4. Proses Verifikasi :

```
>> h1=modulo(v1,R1,p1);
>> h2=modulo(R1,T1,p1);
>> ver1=mod(h1*h2,p1)
ver1 =

    5783
```

Hasil verifikasi cocok

```
>> ver2=modulo(g1,hash1,p1)
ver2 =

    5783
```

```
>> l1=modulo(v2,R2,p2);
>> l2=modulo(R2,T2,p2);
>> veri1=mod(l1*l2,p2)
veri1 =

    12240
```

Hasil verifikasi cocok

```
>> veri2=modulo(g2,hash1,p2)
veri2 =

    12240
```

**Contoh 3.11**

**Proses Verifikasi Dokumen Yang Telah Mengalami Pengubahan**

**1. Menghitung Nilai Hash Dari Dokumen Diterima**

```
>> message2='Dengan ini, diberitahukan bahwa mahasiswa kami yang
bernama Dimas Setya Aji dengan NIM 07305141091 telah menyelesaikan
ujian TA, dan mendapatkan nilai 86,5 dengan indeks A. Demikian telah
dilakukan penilaian secara menyeluruh.';
>> mess1=double(message2);
>> jum=sum(mess1)
jum =

     20221

>> hash=mod(jum,256)+1
hash =

  254
```

**2. Proses Verifikasi Tanda Tangan**

```
>> h1=modulo(v1,R1,p1);
>> h2=modulo(R1,T1,p1);
>> ver1=mod(h1*h2,p1)
ver1 =

    5783

>> ver2=modulo(g1,hash,p1)
ver2 =

   14150

>> l1=modulo(v2,R2,p2);
>> l2=modulo(R2,T2,p2);
>> veri1=mod(l1*l2,p2)
veri1 =

   12240

>> veri2=modulo(g2,hash,p2)
veri2 =

    4128
```

Hasil verifikasi tidak sama

Hasil verifikasi tidak sama